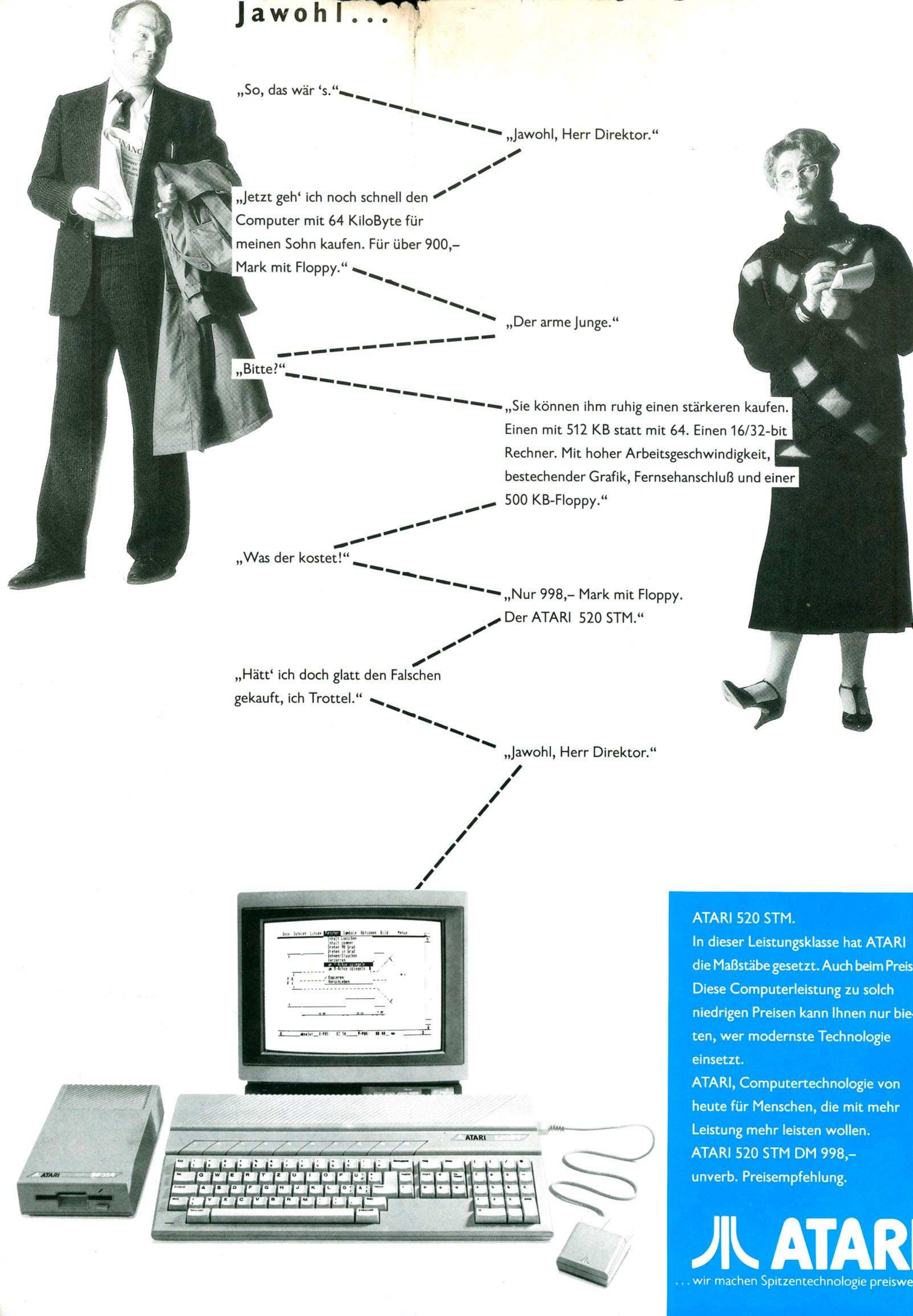
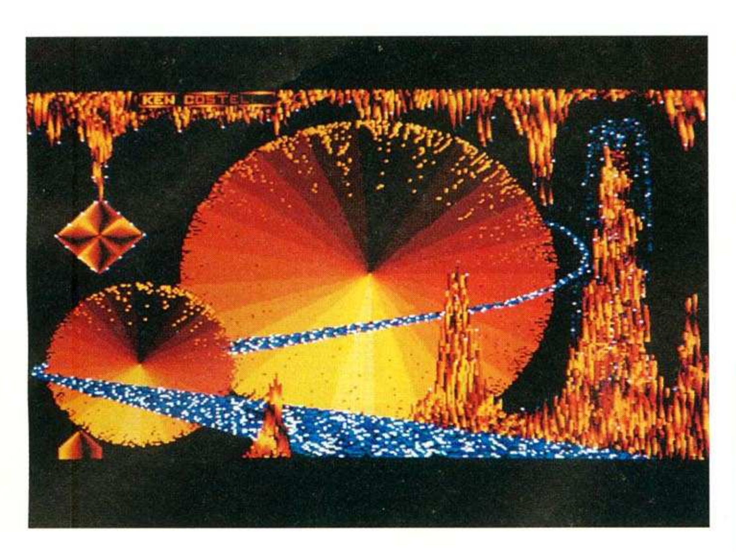
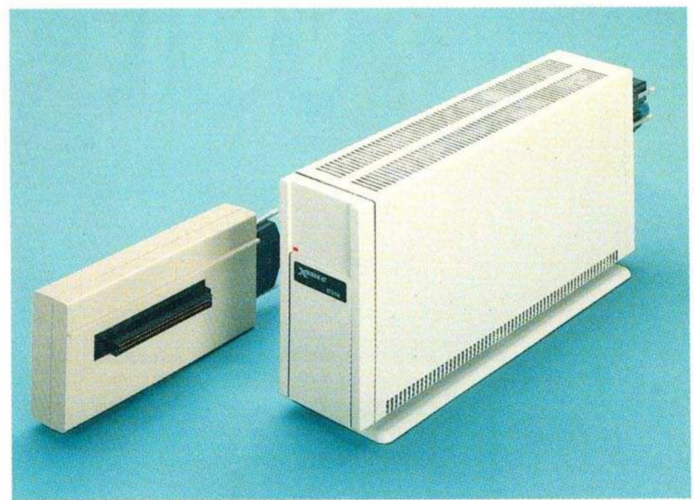

öS 60,–/sfr 7,–/Lit 6.700/hfl 8,50 DM 7,–
AMIGA
Markt&Technik
6/7–'87
DAS COMPUTERMAGAZIN FÜR AMIGA-FANS
Die beste Software für Amiga 500 Amiga 1000 Amiga 2000
Mitmachen: Grafik- und CLI-Kurs
Ideale Drucker für den Amiga
ARD-COMPUTERZEIT
Ergänzende Informationen zu Folge 21 »Robotik«
Erstausgabe


Jawohl...
„So, das wär 's."
„Jawohl, Herr Direktor."
„Jetzt geh' ich noch schnell den Computer mit 64 KiloByte für meinen Sohn kaufen. Für über 900,– Mark mit Floppy."
„Der arme Junge."
„Bitte?"
„Sie können ihm ruhig einen stärkeren kaufen. Einen mit 512 KB statt mit 64. Einen 16/32-bit Rechner. Mit hoher Arbeitsgeschwindigkeit, bestechender Grafik, Fernsehanschluß und einer 500 KB-Floppy."
„Was der kostet!"
„Nur 998,– Mark mit Floppy. Der ATARI 520 STM."
„Hätt' ich doch glatt den Falschen gekauft, ich Trottel."
„Jawohl, Herr Direktor."
ATARI 520 STM.
In dieser Leistungsklasse hat ATARI die Maßstäbe gesetzt. Auch beim Preis. Diese Computerleistung zu solch niedrigen Preisen kann Ihnen nur bieten, wer modernste Technologie einsetzt.
ATARI, Computertechnologie von heute für Menschen, die mit mehr Leistung mehr leisten wollen.
ATARI 520 STM DM 998,–
unverb. Preisempfehlung.
ATARI
... wir machen Spitzentechnologie preiswe



Eine Festplatte für den Amiga . . .

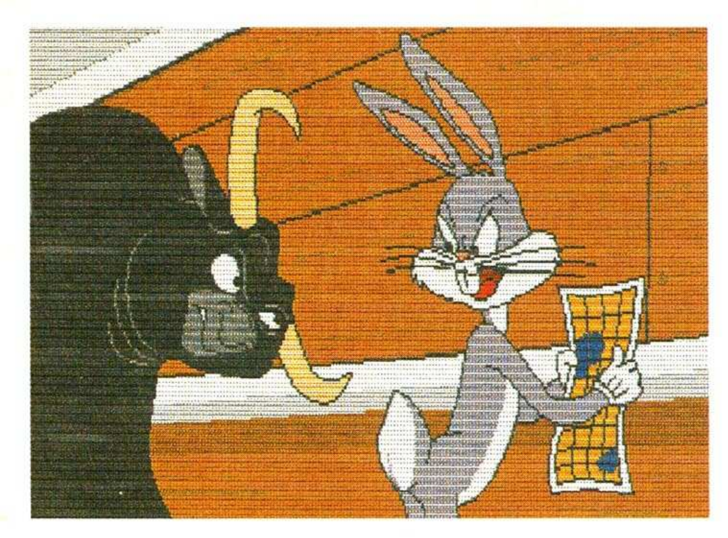
Spätestens, wenn die Speicherkapazität des Computers die der Diskettenlaufwerke übertrifft, sollte sich der Anwender nach einem neuen Speichermedium umsehen. Eine Alternative wäre eine 20-MByte-Festplatte. Wir haben uns drei Geräte für den Amiga angesehen und zeigen Ihnen die Vor- und Nachteile dieser Massenspeicher.
Seite 109



Wunderland Grafik

Der Amiga ist derzeit einer der leistungsfähigsten Grafikcomputer. Doch die Hardware (der Computer) ist nur so gut wie seine Software (die Programme). Deshalb finden Sie in dieser Ausgabe Software-Tests über folgende Grafikprogramme: Delux-Paint II, das derzeit leistungsfähigste Malprogramm, Aegis Images sowie Aegis Draw Plus, ein CAD-Programm mit hoher Leistung.
ab Seite 36

6/7 1987



Amiga — der Spielecomputer

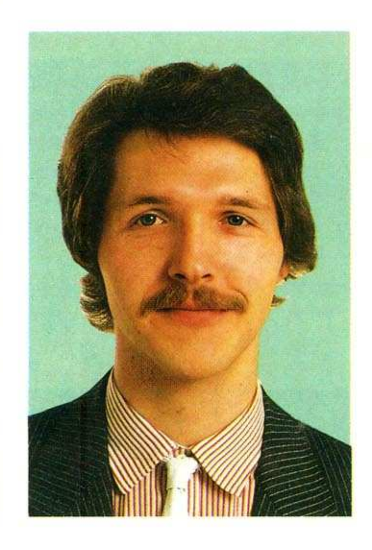
4096 Farben, 4-Kanal-Stereo-Sound und spezielle, superschnelle Grafikchips machen den Amiga zur Spielemaschine par excellence. Deswegen haben im Amiga-Magazin auch Tests ausgewählter Spiele ihren Platz. Getestet wird, was gut ist. Den Anfang machen die Rollen-Spiele »The Bard's Tale« und »Ultima III: Exodus« sowie die Kino-Spiele »Sinbad« und »S.D.I.«. **Seite 118**



Drucker am Amiga

Wir haben vom Laserdrucker bis zum Tinte strahlenden Farbkünstler die wichtigsten Drucker am Amiga ausprobiert und sagen Ihnen, welche Modelle empfehlenswert sind. Damit es beim Anschluß zu keinen Problemen kommt, zeigen wir Ihnen außerdem, wie man sein eigenes Druckerkabel herstellt, das auch mit dem Sidecar funktioniert, und welcher Druckertreiber für Ihren Drucker der richtige ist. **Seite 112**

■ Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind.



Startschuß

Drei erstklassige Rennpferde hat Commodore mit den Amigas im Stall. Der Amiga 500 wird für frischen Wind in der gehobenen Heimcomputerszene sorgen. Mit eingebautem Laufwerk, deutscher Tastatur und fantastischen Grafikeigenschaften eignet er sich geradezu ideal, um als Ein- und Umsteigermodell ein Renner im Computermarkt zu werden. Wer lieber mit abgesetzter Tastatur arbeitet, wird auf den Amiga 1000 zurückgreifen. Für die Anhänger der MS-DOS-Welt bietet sich der Amiga 2000 an. In der Grundversion identisch mit den beiden kleineren Modellen, läßt er sich durch ein ausgeklügeltes Steckkartensystem in einen PC, XT oder AT verwandeln. Alle drei besitzen den 68000-Prozessor und die leistungsfähigen Spezialchips Agnes, Denise und Paula. Gewährleistet soll auch sein, daß ein und dasselbe Programm auf allen drei Versionen lauffähig ist.

Soviel Computer braucht natürlich unterstützende Literatur. Sie halten mit diesem Exemplar die erste Ausgabe der von nun an regelmäßig — zunächst zweimonatlich, dann monatlich — erscheinenden Fachzeitschrift »Amiga« in Händen. Im Amiga-Magazin finden Sie alle Informationen, um mit dem Amiga 500, 1000 bis 2000 sinnvoll arbeiten zu können. Wir werden uns ausschließlich mit dem Amiga beschäftigen, daß heißt Sie bekommen komprimierte Information für Ihren Computer. Allgemeinverständliche Artikel helfen dem mit der Materie noch nicht vertrauten Computerbesitzer und den Einsteigern über die ersten Hürden. Für die Fortgeschrittenen und Profis gibt es von Spezialisten die heißesten Tips und nützliche Routinen — unabhängig von der Programmiersprache. In ausführlichen Kursen werden Sie mehr über bestimmte Teilbereiche beispielsweise die Grafikprogrammierung, CLI-Anwendung oder Sprachen wie Basic, C und Pascal erfahren.

Umfassende Testberichte sowohl über neue Software- als auch über Hardware-Produkte geben Ihnen die nötigen Grundlagen zur Hand, um sich für das richtige Programm oder den passenden Drucker zu entscheiden. Das Angebot an Grafik-, Textverarbeitungs-, Musik- oder Kalkulationsprogrammen wächst ständig; die verschiedenen Drucker am Markt lassen sich kaum noch zählen; sorgfältig zusammengestellte Marktübersichten im Amiga-Magazin erleichtern Ihnen die Orientierung. Sie sind an Top-News rund um den Amiga interessiert, wollen über Trends und Tendenzen Bescheid wissen, mehr über die »Cracks« in der Szene erfahren — mit dem aktuellen Teil des Amiga-Magazins sind Sie immer dabei.

Dabei sein sollten Sie auch in der aktiven Mitgestaltung des Amiga-Magazins. Konstruktive Kritik ist genauso willkommen wie gute Listings, Testberichte oder gar komplette Kurse. Der Startschuß ist gefallen; Sie als Leser bestimmen mit, wie nutzbringend Ihre Computerzeitschrift für Sie ist.

Ihr
Albert Absmeier
Chefredakteur

68000er

Die **68000er-Ausgabe 6** bietet wieder brandaktuelle Information rund um den Amiga.

Sollte Ihr Amiga 1000 am Videorecorder oder Videomonitor nur ein Schwarzweiß-Bild liefern, dann öffnet unser Basteltip der Juni-Ausgabe die Tore zur Farbenwelt.

Verblüffende Soundeffekte und fetzige Musikstücke packen Sie mit unserer Steuersoftware zum Amiga-Digitizer in Ihren Computer. Bis zu einem halben Megabyte Musik digitalisieren Sie damit schnell und komfortabel.

Die Desktop Publishing-Welle rollt nun auch unaufhaltsam auf den Amiga zu. Wie einfach ist es, mit Pagesetter eine Druckseite zusammenzustellen und auf Papier zu bannen?

Und bei unserer Leserumfrage können Sie eine Reise zum Hightech-Dorado an die amerikanische Westküste gewinnen! Stationen der Reise: Las Vegas, Sunnyvale, San Francisco. Also mitgemacht beim Super-Wettbewerb!

Die **Ausgabe 7 unseres 68000er-Magazins** räumt auf in der Gerüchteküche. Wir sagen Ihnen klipp und klar, welche der bekanntesten und wichtigsten Programme auf dem Amiga 500 laufen und welche nicht.

Eine neue Dimension in der Computermusik erschließt Sonix, ein brandneues Musikprogramm für den Amiga. Ob mitreißende Trommelwirbel, Synthesizersoli oder die Steuerung von MIDI-Instrumenten: Das Allroundtalent Sonix meistert alle Musik-Disziplinen.

Wenn ein Videorecorder oder ein anderer Computer am Videoeingang des 1081-Monitors angeschlossen wird, so stört der eingeschaltete Amiga das Videobild enorm. Zwei Drähte lösen das Problem.

Dem Geheimnis der Amiga-Icons gehen wir in Theorie und Praxis auf den Grund. Ein C-Programm wandelt beliebig große Deluxe Paint-Brushes in Icons um, die nach dem Anklicken auch eine andere Gestalt annehmen können.



Die **Ausgabe 6/87** der Happy-Computer beschäftigt sich mit dem Thema Grafik und Computer. Der Beuys-Schüler Tom A. Hawk zeigt, wie man mit dem Amiga und Deluxe Paint Kunstwerke malt. Tolle Preise winken bei unserer Leserumfrage. Sagen Sie uns Ihre Meinung und gewinnen Sie einen Amiga 500 oder einen anderen von über 750 Preisen. Mitmachen lohnt sich!

Als Amiga-Listing gibt es eine Software-Spezialität: Ein selbstlernendes Vier-gewinnt-Programm macht den Amiga zum superstarken Gegner. Wer sich für die Welt der Fractals interessiert, findet ein Listing, um sich in einer künstlichen Fraktal-Landschaft mit See zu bewegen.

Im großen Spiele-Sonderteil stehen wieder die Tests der neuesten Programme. Wir stellen zum Beispiel das brandaktuelle Adventure »Guild of Thiefs« der Programmierer von »The Pawn« vor. Wer bei The Bard's Tale verzweifelt, findet die Lösung in dieser Ausgabe.

Mitte Juni erscheint die **Ausgabe 7/87** der Happy-Computer, die sich eingehend mit Massenspeichern beschäftigt. Wohin geht der Tend? Wie sieht die Zukunft aus? Hier erfahren Sie alles über Festplatten, was sie kosten und wie sie funktionieren. Im Test: ein echtes CD-ROM mit über 600 MByte Speicherkapazität. Das ist genug Platz für mehr als 700 Public Domain-Disketten und eine komplette Bibel auf einem CD-ROM. Ist das der Speicher von morgen?

Der Grundlagenteil erklärt, wie leistungsfähige Packer funktionieren und wie man sie selbst programmiert. So sparen Sie viel Platz auf Ihren teuren Disketten.

Was ist los bei der »Konkurrenz«? Ein Schwerpunkt dieser Ausgabe widmet sich Atari. Wie fing alles an? Wer sind die Entwickler aus Kalifornien? Was macht das alte Commodore-Team bei Atari? Kann Tramiel kontern? Viele Fragen in einem Spiel mit vielen Unbekannten.

Kurztest: Vizawrite und Prowrite für den Amiga

Zwei Textverarbeitungen für den Amiga streiten um den ersten Platz — wir haben uns Vorversionen von Prowrite und Vizawrite-Amiga angesehen und einem kritischen Vergleichstest unterzogen.

Prowrite ist ein Produkt der US-Firma New Horizons und wird mit Unterstützung der Schweizer Softwareschmiede AmiLink auf die Erfordernisse des europäischen Marktes angepaßt. Vizawrite Amiga ist die Amiga-Version des bereits auf dem C 64 und IBM-PC erfolgreichen Vizawrite aus dem britischen Softwarehaus Viza Software. Beide Produkte unterstützen die Benutzeroberfläche Intuition durch Pull-Down-Menüs, Fenster und Dialogboxen. Vizawrite setzt auf Benutzerfreundlichkeit und arbeitet nur mit grafischer Benutzeroberfläche (Bild 1). Prowrite (Bild 2) bietet zusätzlich noch die Bedienung über Schlüsseltasten; für Daueranwender bringt das einen Geschwindigkeitsvorteil und zudem eine Steigerung der Leistungsvielfalt ohne zeitraubende Pull-Down-Menüs. Für den Gelegenheitsanwender ist jedoch die Arbeit mit Pull-Down-Menüs einfacher.

Ein Editieren des Textes mit Cut- und Paste-Funktionen (einfach Text mit der Maus aus dem Bild ausschneiden und dann reformatieren, löschen oder kopieren) ist mit beiden Programmen einfach und problemlos zu handhaben. Während sich Vizawrite der europäischen PAL-Norm mit einer vertikalen Auflösung von 256 Punkten anpaßt, geht Prowrite noch einen Schritt weiter: Im niedrigauflösenden Modus stehen 281 Rasterzeilen für den Text zur Verfügung, im Interlace-Modus sogar das Doppelte. Vizawrite arbeitet vertikal mit 320 Punkten (Lores), Prowrite stellt 640 Punkte dar, eine Erweiterung auf 690 Punkte ist geplant. Ist der Text länger als eine Bildschirmseite, wird bei beiden Textprogrammen das Bildschirmfenster nach oben oder unten verschoben (gescrollt) — bei Prowrite ist das relativ langsam, dafür aber nicht so oft notwendig wie beim schnelleren Vizawrite. Auch bei der Texteingabe zeigt sich Vizawrite etwas schneller als Prowrite, bietet aber wegen der geringen Auflösung kein so intensives »What you see is what you get«-Gefühl. Das ständige Textscrollen bei Vizawrite macht also den Geschwindigkeitsnachteil von Prowrite wieder wett.

Hohe Leistungsvielfalt

Wie vom deutschen Viza-Vertrieb DTM zu hören war, bastelt Kevin Lacy allerdings gerade an einer schnelleren Vizawrite-Version. Beide Programme erlauben eine Einbindung von Bildern im IFF-Format (DeluxePaint, Graphicraft, Aegis Images) in den Proportionalschrift-Text. Vizawrite hält sich dabei eng an das Vorbild Macwrite — Texte können nur über und unter dem Bild dargestellt werden, aber nicht daneben. Auch hier ist Prowrite dem britischen Konkurrenten einen Schritt voraus: Texte werden auf Wunsch um das Bild herum zentriert. Bilder für Vizawrite müssen in der derzeitigen Version noch schwarzweiß (1 Bitplane) sein, während das schweizerisch-amerikanische Produkt beliebige Bilder laden, in allen Farben darstellen und auch drucken kann. Sowohl Vizawrite als auch Prowrite arbeiten mit beliebigen Schriftarten aus dem Font-Directory; wer einmal wissenschaftliche Notationen schreiben will, benötigt also keine neue Textverarbeitung, sondern nur einen Fonteditor oder die von Fremdherstellern vertriebenen fertigen Zeichensätze. Die Zeichen können mit Vizawrite in Schwarzweiß dargestellt werden, während Prowrite mit bis zu sieben Textfarben arbeitet und auch die mit dem Fontprogramm »Calligrapher« erstellten Color-Fonts übernimmt. Die Zeichensätze und auch der Font-Stil (normal, fett, kursiv, unterstrichen, hoch- und tiefgestellt) können beliebig im Text Buchstabe für Buchstabe geändert werden. Der Zeilenabstand wird automatisch an die Größe der Zeichensätze angepaßt — so intelligent sind beide Programme. Prowrite bietet zusätzlich noch die Einstellung eines doppelten Zeilenabstandes.

Die technische Leistungsfähigkeit beider Programme ist — abgesehen von Farben, Auflösung und Geschwindigkeit — in etwa die gleiche. So kann

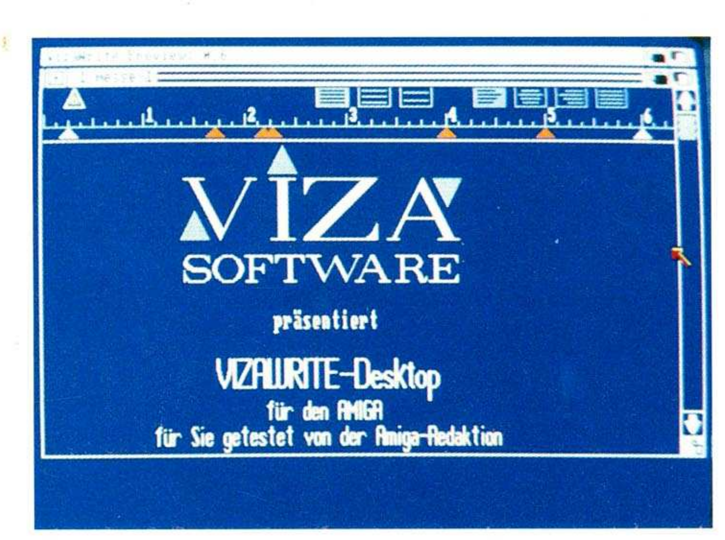


Bild 1. Vizawrite Amiga — ein vertrauter Bekannter mit neuem Gesicht

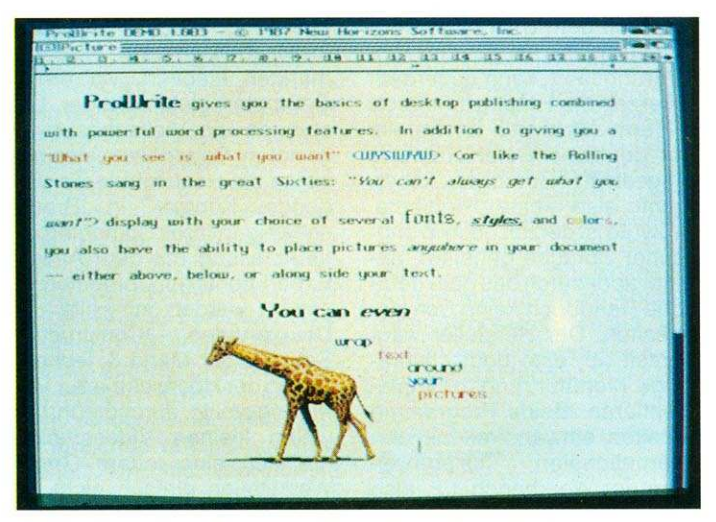


Bild 2. Prowrite gibt dem Anwender echtes »What you see is what you get«

man beispielsweise beliebig viele Texte (beschränkt nur durch den Speicher) gleichzeitig in verschiedenen Windows bearbeiten. Beide Textprogramme arbeiten mit Proportionalschrift, die wahlweise linksbündig, rechtsbündig, zentriert oder im Blocksatz dargestellt werden kann. Die in Vizawrite vorhandene Möglichkeit, die Funktionstasten mit Floskeln zu belegen, wurde in Prowrite nicht implementiert. Durch die etwas größere Funktionsvielfalt von Prowrite kann man über diesen Minuspunkt allerdings hinwegsehen. Prowrite wird in der Endversion voraussichtlich mit einem neuen Druckertreiber ausgeliefert, der (gegen alle Gewohnheiten der verhaßten Systemdruckertreiber) die Befehle verschiedener Drucker voll nutzen kann. Vizawrite-Amiga kann wahlweise mit den in Preferences eingestellten Systemdruckertreibern arbeiten, gestattet aber außerdem, einen zusätzlichen neuen Druckertreiber einzusetzen. Wer mit diesem Treiber arbeitet, benötigt zwei Zeichensätze: einen für den Bildschirm, einen für den Drucker.

Interlace-Modus — zum Schreiben geeignet?

Die Arbeit mit Prowrite im Interlace-Modus ist für die Augen ermüdend, da man die hellen Hintergrundfarben leider nicht ändern kann. Der Hersteller empfiehlt daher seinen Monitoraufsatz »Chitter Reed«, der das Interlace-Flimmern stark vermindern soll. In Deutschland ist eine Auslieferung mit der Anti-Flicker-Scheibe geplant. Das Tool »Anytime« von Discovery Software (wird mit dem Utility »Grabbit« geliefert), das als Task im Hintergrund läuft, kann außerdem jederzeit die Hintergrundfarbe ändern. Vizawrite ist bereits auf deutsche Befehlsführung umgestellt; Benutzer haben damit also keine Probleme. Prowrite ist derzeit noch in englischer Sprache gehalten, bereitet aber durch das gute deutsche Handbuch keine Schwierigkeiten. Der Hersteller wägt derzeit ab, eine deutschsprachige Menüführung zu implementieren. Beide Programme arbeiten einwandfrei mit den internationalen Tastaturtreibern der Workbench 1.2, also auch mit deutscher Tastatur. Die Druckertreiber passen sich dabei an den Tastaturtreiber an. Insgesamt gesehen sind sowohl Vizawrite als auch Prowrite sehr leistungsfähige Produkte, mit denen sich professionell arbeiten läßt. Prowrite bietet ein größeres Leistungsspektrum, Vizawrite überzeugt durch hohe Geschwindigkeit und einfachere Bedienung. Wer das Rennen um den ersten Platz gewinnen wird, ist derzeit noch nicht abzusehen. Im harten Konkurrenzkampf arbeiten beide Firmen an ständigen Verbesserungen der Programme. In den derzeit aktuellen Versionen überzeugte Prowrite durch seinen günstigeren Preis. Wenn die Endversion von Vizawrite-Amiga auf dem Markt ist, wird man mit 498 Mark dabei sein. Prowrite wird voraussichtlich für etwa 350 Mark zu haben sein. (Manfred Kohlen/sk)

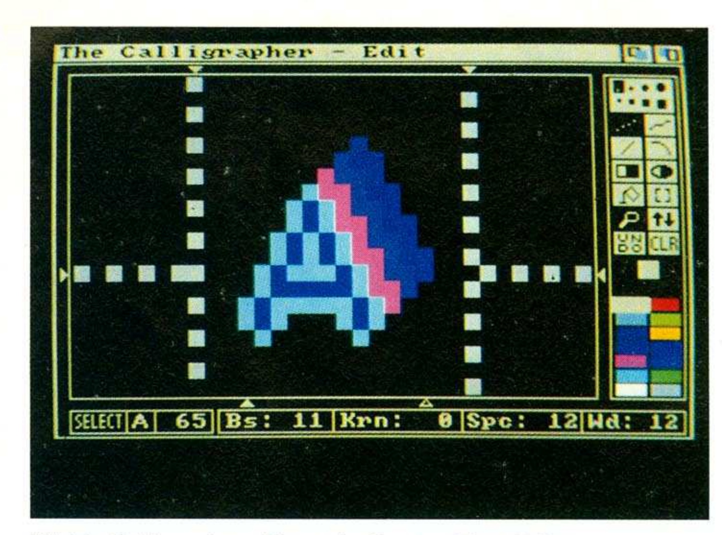


Bild 3. Calligrapher: Einer der besten Font-Editoren

Der Amiga in Nischenmärkten

Ganz wie von Commodore vor etwa eineinhalb Jahren vorausgesagt, hat sich der Amiga inzwischen in den Nischenmärkten etabliert, wo bisher Computer gleichwertiger Leistungen ein Vielfaches mehr kosteten. So arbeitet beispielsweise die Hochschule für »Bildende Künste« in Braunschweig in ihrer Computergrafik-Abteilung mit dem Amiga. Experimente mit Computeranimation werden mit Hilfe des Deluxe-Video Construction Set, das der Markt & Technik-Verlag der Hochschule zur Verfügung stellte, durchgeführt. In vielen kleinen Videostudios, die sich eine teuere Grafik-Workstation einfach nicht leisten können, wird der Amiga inzwischen ebenfalls eingesetzt. Die üblichen Erweiterungen dazu sind ein Genlock-Interface, Grafik-Digitizer, Malprogramme wie Deluxe-Paint II und Animationsprogramme wie Deluxe-Video und Aegis Animator.

Auch in der Forschung hat der Amiga seinen Platz gefunden: Das in Genf ansässige Kernforschungsinstitut CERN (mit über 6000 beschäftigten Wissenschaftlern!), das gerade einen neuen Atombeschleunigungsring aufbaut, hat sich ebenfalls einen Amiga angeschafft. Die Rechenzeiten der VAX-Rechner sind ausgebucht, die Cray II kommt erst in einigen Monaten — also mußte man sich für etwas »Kleines« als Zwischenschritt entscheiden. Nach genauer Abwägung und Tests entschloß man sich schließlich für den Amiga. Bei Benchmark-Tests für die nötigen Berechnungen schnitt der Amiga in der Geschwindigkeit am besten ab.

Erst jetzt, nach der Etablierung des Amiga in solchen Märkten, dringt Commdore mit dem Amiga 500 in den Heimbereich vor — Doch schon zuvor hat der Amiga sich den Platz in Randmärkten wie Forschung und Profi-Animation erobert. (Manfred Kohlen/sk)

Neue Software für den Amiga

Durch den Amiga 500 wachgerüttelt, scheinen nun fast alle Softwarehersteller auf Amiga-Kurs zu schwenken. Hersteller, die sich bereits vorher dem Amiga 1000 verschworen hatten, können bereits jetzt mit qualitativ ausgereiften Produkten aufwarten.

Die deutsche Firma DTM will den in Eigenproduktion entstandenen Diskettenmonitor und -doktor »Discovery« im Laufe der nächsten Wochen auf den Markt bringen. Das für 198 Mark erhältliche Produkt überzeugt durch unzählige Funktionen, einfache Bedienung und ausgesprochen hohe Geschwindigkeit. Ständige Updates für neue Erfordernisse sind geplant. Ein ausführlicher Test des Programms wird folgen.

Die schweizerisch-amerikanische Softwaregruppe AmiLink entstand aus kleinen Softwarefirmen und freien Programmierern, um durch zueinander kompatible Softwareprodukte den Anwender zu unterstützen und die Produkte den Erfordernissen der internationalen Märkte besser anpassen zu können. So darf man in Zukunft einige interessante Produkte aus dieser Softwareschmiede erwarten. »Calligrapher«, ein Programm zum Erstellen von Zeichensätzen, bietet neben einem Fonteditor einige recht außergewöhnliche Leistungsmerkmale (Bild 3). So können beispielsweise Ecken an Buchstaben durch einen einzigen Mausklick abgerundet werden, ganze Zeichensätze beliebig vergrößert und verkleinert oder sogar verbogen werden. Will man eine Schrift mit Schatten erzeugen, verschiebt man einfach die Lichtquelle mit der Maus, und schon ist der Schatten für den gesamten Zeichensatz angepaßt. Zusatzbibliotheken bringen weitere Effekte mit sich (zum Beispiel Helligkeitsveränderungen innerhalb der Buchstaben). Die — auch farbigen — Zeichensätze lassen sich in das Standard-Font-Directory kopieren und für andere Produkte weiterverwenden. So arbeitet beispielsweise die Textverarbeitung »Prowrite« mit diesen Zeichensätzen, sie können aber auch für das Hold-and-Modify-Zeichenprogramm »Prism« oder das Effektprogramm »Showmaker« verwendet werden.

Beide Produkte sind übrigens ebenfalls Ergebnisse dieser internationalen Zusammenarbeit. »Prism« ist ein Malprogramm in der Güteklasse eines DeluxePaint, arbeitet aber mit 4096 Farben gleichzeitig. Beliebige IFF-Bilder lassen sich in Prism einladen und zu neuen Bildern zusammensetzen — so kann man die Leistungsmerkmale auch anderer Grafiksoftware anwenden, ohne jedoch auf 32 Farben beschränkt zu sein. Die zusammengesetzten Bilder können

in Prism dann weiterverarbeitet werden.

»Showmaker« ist gerade im Entstehen. Es handelt sich um eine Art Vorführprogramm. Da fliegt zum Beispiel ein IFF-Bild langsam von vorne nach hinten (Star Wars Titelschrift-Effekt), stellt sich gerade auf dem Bildschirm auf; kurz darauf saust ein Stein in das Bild, und mit akustischer Untermalung fallen die Einzelteile des Bildes als Scherben nach unten. (Manfred Kohlen/sk)

Präsident jetzt auch für Amiga

Laut Auskunft des Importeurs wird der Drucker »Präsident 6313« neuerdings mit dem Commodore Amiga- oder dem Atari ST-Zeichensatz angeboten. Durch das modulare Schnittstellenkonzept ist der Präsident somit an nahezu jeden gebräuchlichen Computer anzupassen. Unverbindliche Preisempfehlung: 498 Mark. (Oliver Reichert/pd)

Horst Grubert, Import und Agentur, 8110 Murnau-Riegsee, Tel. (08841) 8011

Turbo-Basic für den Amiga 2000

War Basic als Programmiersprache in den Kreisen der PC- und 16-Bit-Programmierer langsam dabei in Verruf zu kommen, so ändert sich dies mit einem neuen Produkt von Heimsoeth & Borland gewaltig. Turbo-Basic ist nach Turbo-Pascal und Turbo-Prolog die dritte Programmiersprache aus dem Hause Borland und kombiniert die Stärken von Pascal und Basic. Das Produkt ist sowohl vom Inhalt, als auch von der äußeren Form her sehr ansprechend aufgemacht und bietet ein komplettes Basic-Entwicklungssystem für IBM-PCs und Kompatible, wie ihn der Amiga 2000 auch darstellt. Es beinhaltet einen »Memory to Memory Compiler«, einen Full-Screen-Editor, einen internen Linker und eine Runtime-Bibliothek. Wie die anderen Borland-Produkte arbeitet auch Turbo-Basic mit Pull-Down-Menüs und ist somit sehr benutzerfreundlich.

Der erzeugte Maschinencode glänzt durch Geschwindigkeit, gestattet strukturiertes Programmieren und adressiert volle 640 KByte Speicher. (sk)

Turbo-Basic, empf. Verk.-Preis 250 Mark, Heimsoeth & Borland, Postfach 140280, 8000 München 5

Der dritte im Bunde: Scribble!

Herrschte auf dem Markt der Amiga-Software in Sachen Textverarbeitung mehr oder weniger Ebbe, so kommt nun auch hier einiges in Bewegung. Wir haben uns »Scribble!« von Micro-Systems Software Inc. für Sie angesehen.

Neben der Tabellenkalkulation »Analyze!« und der Datenbank »Organize!« bietet das amerikanische Software-Haus Micro-Systems Inc. die Textverarbeitung »Scribble!« für den Amiga an.

Auch bei diesem Programm setzt sich eine sehr begrüßenswerte Tendenz von Seiten der Software-Anbieter durch: Scribble! ist nicht kopiergeschützt und auf den ersten Seiten des Handbuches wird dem Anwender empfohlen, sich zunächst eine Arbeitskopie der Originaldiskette anzulegen.

Nach dem Starten des Programms, das sowohl auf dem Amiga 500 als auch auf dem Amiga 1000 ohne Probleme läuft, befindet man sich sofort im Editier-Modus. Scribble arbeitet im normalen 80-Zeichen-Modus des Amiga und erfordert somit kein spezielles Zubehör wie Filterscheiben oder ähnliches (bei anderen Programmen dank des häufig verwendeten Interlace-Modus durchaus nichts Ungewöhnliches). Der Mauszeiger hat die Form eines Bleistifts, welche sich allerdings bei Anwahl anderer Programmfunktionen entsprechend ändert. Die Steuerung von Scribble! entspricht ganz der Philosophie der Amiga-Software und funktioniert über Pull-Down-Menüs. Allerdings können die wichtigsten und am häufigsten gebrauchten Editier- und Schreibfunktionen auch über die Funktions-Tasten oder Kombinationen mit der ALT- oder der Amiga-Taste erreicht werden. Hierdurch wird die Bedienung des Programms um einiges schneller. Eine weitere sehr angenehme Eigenschaft des Programms sind zwei über <F1> und <F2> abrufbare »Help«-Seiten, die einen Überblick über die wichtigsten Kommandos und die entsprechenden Tastenkombinationen geben.

Durch die hervorragenden Grafikfähigkeiten des Amiga läßt sich die Palette an Textverarbeitungsprogrammen in zwei Kategorien aufteilen. Hier sind zum einen die sogenannten »WYSIWYG«-Programme (What you see ist what you get), die den Text inklusive den eingebundenen Grafiken so auf dem Bildschirm darstellen, wie er später gedruckt erscheint (sofern der Drucker dies beherrscht). Zum anderen gibt es die Programme herkömmlichen Stils. Hier wird der Text zwar auf dem Bildschirm formatiert, spezielle Druckattribute werden jedoch erst beim Ausdruck sichtbar. Scribble! liegt etwa in der Mitte zwischen beiden Kategorien. Es werden zwar Unterstreichungen, »Bold«- und Kursiv-Schrift am Monitor angezeigt, andere Veränderungen des Textes hinsichtlich des Druckbildes werden jedoch nicht dargestellt. Hier muß für Sub- und Superscript gar mit Steuerzeichen im Text gearbeitet werden. Dies mag dem grafikverwöhnten Amiga-User zwar etwas antiquiert vorkommen. Die Bearbeitungsgeschwindigkeit des Amiga und das Scrolling durch den Text geht hierdurch aber sehr flott vonstatten. Eine Grafikeinbindung in den Text ist ebenfalls nicht vorgesehen. Doch das Verwenden von Grafiken kommt ohnehin seltener vor, als es das Gewicht, das in der Werbung oft auf diese Möglichkeit gelegt wird, vermuten läßt. Etwas unangenehmer fallen hier schon Steuerzeichen im Text auf, die auf einem solchen Computer nicht mehr unbedingt der Höhe der Zeit entsprechen. Als logische Konsequenz aus dem obengenannten bietet Scribble natürlich einen »Preview«-Modus an, in welchem der Text dem späteren Druckbild entsprechend am Monitor erscheint.

Dies zu den speziellen Amiga-Features, die man bei Scribble! zum Teil vermißt. Das soll jedoch nicht darüber hinwegtäuschen, daß Scribble! ansonsten ein grundsolides Textverarbeitungsprogramm darstellt. Es sind alle Funktionen wie Suchen und Ersetzen, Blockoperationen etc. vorhanden und arbeiten auch problemlos. Eine Serienbrief-Funktion, die man mit Fug und Recht zum Standard zählen kann, bietet Scribble! natürlich auch. Sehr schön ist auch die »Spell-Check«-Funktion. Diese kann direkt von der Menüleiste aus aufgerufen werden und bietet schon in der ausgelieferten Version einen recht umfangreichen englischen Wortschatz. Und hier liegt der Haken des Programms, den es sich zur Zeit noch mit vielen anderen Textprogrammen für den Amiga teilt:

Es wird geliefert mit englischem Handbuch, besitzt englische Bedienerführung und verfügt über keine deutschen Umlaute. Laut Auskunft von Micro-Systems ist eine deutsche Version in Arbeit und soll bald erscheinen. (sk)

Atlantis, In der Henn 27, 5030 Hürth 3, 228 Mark
Micro-Systems Software Inc., 4301-18 Oak Circle, Boca Raton, Florida 33431
Interplan, Nymphenburger Str. 134, 8000 München 19, 330 Mark

Diskettenmonitor mit deutschem Handbuch

Seit kurzem wird von Atlantis ein Diskettenmonitor angeboten, mit dem man die Organisation der Amiga-Disketten ergründen kann. Im Handbuch wird dabei nicht nur auf die Funktionsweise des Produkts eingegangen, sondern auch auf den Diskettenaufbau selbst. So wird dem Anwender der Umgang mit der Dateiorganisation auf der Diskette erheblich erleichtert. Daneben lassen sich, wie von einem professionellen Produkt nicht anders zu erwarten, gelöschte Dateien zurückgewinnen, Fehler auf der Diskette korrigieren oder Daten aus beschädigten Dateien retten. Einzelne Sektoren können am Bildschirm Hexadezimal oder in ASCII-Darstellung editiert werden. Gesteuert wird das Programm mit Hilfe der Maus. Der Preis beträgt inklusive deutscher Anleitung 108 Mark. (rf)

Atlantis, In der Henn 27, D-5030 Hürth 3, Diskwik, 108 Mark

Pascal nach ISO

Der von Metacomco angebotene Pascal-Compiler hält sich voll und ganz an den ISO-Standard und bietet darüber hinaus noch Zusätze zum Einbinden von externen Programm-Modulen, sowie eine Funktion zum Nachladen von Programmen. Für die Dateiverwaltung wurde MCC-Pascal um die Prozeduren WRITE und REWRITE erweitert. Leider fehlen aber integrierte Funktionen, um die grafischen und musikalischen Fähigkeiten des Amiga ausnutzen zu können. Das Programm kostet zwischen 238 und 249 Mark. Ein englisches Handbuch liegt bei. (rf)

Atlantis, In der Henn 27, 5030 Hürth 3, MCC-Pascal, 238 Mark
Philgerma, Ungererstr. 42, 8000 München 40, MCC-Pascal, 248 Mark
Softwareland, Franklinstr. 27, CH-8050 Zürich, MCC-Pascal, 249 Mark
Interplan, Nymphenburger Str. 134, 8000 München 19, 398 Mark

Lisp: »Intelligenz« für den Amiga

Immer wieder erhitzt ein Thema die Gemüter, über das sich selbst die meisten Experten nicht so ganz einig sind: KI (Künstliche Intelligenz). Ein Begriff, dessen Definition wegen der Komplexität des Themengebietes noch immer Schwierigkeiten bereitet. Kann man nun besonders »intelligente« Anwenderprogramme, wie etwa Desktop-Publishing bereits als KI bezeichnen oder muß hier mit einer speziellen Sprache wie Lisp oder Prolog vorgegangen werden? Die verwendeten Sprachen spielen hier sicherlich eine große Rolle, wie Sie gleich sehen werden. Jeder KI-Experte wird sich allerdings weigern ein besonders gut gelungenes Anwendungs-Programm der KI zuzuordnen. Der KI-Freund fühlt sich prinzipiell zu höherem berufen. Er verwirklicht mit Vorliebe Expertensysteme, Bilderkennungssysteme und andere teilweise noch utopische Projekte. Utopisch deshalb, weil die KI-Forschung trotz intensivster Bemühungen noch nicht so recht den Kinderschuhen der Grundlagenforschung entfliehen konnte. Intelligenz bedeutet im Allgemeinen, daß auf Einwirkungen von außen durch spezielle Erfahrungen reagiert wird. Als »Erfahrung« zählt hier auch das Wissen, sich bei Regen mit Hilfe eines Regenschirms vor Nässe zu schützen. Versuchen Sie doch einmal diesen Sachverhalt, neben vielen anderen »Lebenserfahrungen« auch, Ihrem Computer beizubringen. Mit dem normalen Basic eine endlose Kette von IF-Abfragen, deren Umfang bald den vorhandenen Speicherplatz sprengen würde. Überhaupt stellt der Speicherplatz eines der größten Probleme der KI dar. Um diesem Manko zu begegnen, wurden nun spezielle Programmiersprachen geschaffen, die nicht mehr einfach Befehlsketten abarbeiten, sondern in der Lage sind, eine vorgegebene Wissensbasis nach bestimmten Regeln zu durchsuchen oder zu verknüpfen. Eine diese Sprachen ist das bereits Eingangs erwähnte Lisp. Diese KI-tüchtige Sprache steht schon seit längerer Zeit auch für den Amiga zur Verfügung. Metacomco zeichnet für einen Lisp-Interpreter/Compiler verantwortlich, der dem Amiga einen Hauch von Intelligenz verleihen soll.

Wer als Neuling in die KI-Problematik einsteigen will, wird allerdings gehörige Schwierigkeiten haben, mit diesem Produkt umzugehen. Die Anleitung ist komplett in Englisch verfaßt, aber mit normalem Schulenglisch durchaus lesbar gestaltet. Als Lektüre für Lernzwecke allerdings ist das beigelegte Handbuch ungeeignet. Man muß also auf Zusatzliteratur ausweichen. Der Markt in diesem Bereich ist aber, was deutschsprachige Literatur angelangt, noch nicht sehr ergiebig.

Wie auch bei herkömmlichen Programmiersprachen gibt es für Lisp einen Standard, an den sich ein Interpreter/Compiler nach Möglichkeit halten sollte: Cambridge Lisp und Common Lisp. Metacomco wählte für seine Lisp-Version den Cambridge-Standard, der sich von seinem Artverwandten allerdings nur unwesentlich unterscheidet. Einige kleine Änderungen sind im Handbuch beschrieben, so daß der Lisp-Kenner hiermit kaum Probleme hat. Erfaßt werden die Programme innerhalb der Lisp-Umgebung, so daß sich diese auch sofort austesten lassen. Daneben lassen sich noch verschiedene »Fehlerfallen« einbauen, die eine komfortable Fehlersuche ermöglichen. Vor allem bei der Verwendung von den in der KI so wichtigen Binärbäumen, eine unentbehrliche Hilfe für den Programmierer.

Alles in allem präsentiert sich das Metacomco-Cambridge-Lisp als vollwertige Hilfe für die KI-Programmierung mit dem Amiga. Zu beachten ist allerdings noch, daß die von uns getestete Version nur unter Kickstart 1.1 läuft. Besitzer eines Amiga 500 können den Interpreter/Compiler deshalb noch nicht nutzen — ein Problem, das leider nicht nur dieses Produkt betrifft.

So gut die Software auch sein mag, der Preis jedenfalls dürfte beim Kauf ein großes Hindernis darstellen. Cambridge Lisp wird zwischen 378 Mark und 490 Mark verkauft. Es bleibt zu hoffen, daß sich hier noch einiges ändern wird, um diese Sprache auch für den Einsteiger interessant zu machen. (rf)

Interplan, Nymphenburger Str. 134, 8000 München 19, 598 Mark
Atlantis, In der Henn 27, 5030 Hürth 3, Metacomco Cambridge Lisp, 378 Mark
Philgerma, Ungererstr. 42, 8000 München 40, Metacomco Cambridge Lisp, 490 Mark
Softwareland, Franklinstr. 27, CH-8050 Zürich, Metacomco Cambridge Lisp, 390 Mark

Dateiverwaltung von MSS

Organize!, ein leistungsfähiges Dateiverwaltungsprogramm von Micro-Systems-Software Inc., steht jetzt auch in Deutschland zur Verfügung.

In der uns vorliegenden Version 1.0 kann das Programm theoretisch die beachtliche Menge von 4,2 Milliarden Datensätzen verwalten — eine Zahl, die durch die Größe der Datensätze und die Kapazität des Massenspeichers allerdings eingeschränkt wird. Jeder Datensatz kann dabei aus bis zu 128 Feldern zu je 254 Zeichen bestehen, was für die meisten Anwendungen wohl vollauf genügen dürfte.

An leistungsfähige Sortierfunktionen wurde ebenso gedacht wie an eine einfach zu bedienende Benutzeroberfläche, die sich an Intuition orientiert (Bild 1). Alle Funktionen können mit Pull-down-Menüs aufgerufen werden.

Das nicht kopiergeschützte Programm wird auf einer 3½-Zoll-Diskette geliefert und setzt einen Amiga mit mindestens 512 KByte RAM und einem Diskettenlaufwerk voraus. Die Installation auf einer Festplatte bereitet jedoch keinerlei Schwierigkeiten.

Das ausführliche Handbuch gibt — zur Zeit noch in Englisch — über jede Funktion hinreichend Auskunft.

Ein besonderer Leckerbissen ist die Tatsache, daß dBASE- und ASCII-Dateien eingelesen werden können. Somit wird dem Anwender auch das Umsteigen auf Organize! denkbar leicht gemacht.

Micro-Systems bietet den amerikanischen Anwendern übrigens (falls notwendig) umfangreiche Hilfestellung: Für das Programm existiert eine eigene Hotline sowie eine firmeneigene Mailbox und es gibt zudem die Möglichkeit, über CompuServe — eine der größten Datenbanken der USA — Nachrichten und Hilferufe an den Hersteller zu senden. Ein Beispiel, das auch bei uns Schule machen sollte.

In Deutschland wird das Programm von Atlantis und Softwareland vertrieben, der Verkaufspreis liegt bei knapp 230 Mark. (pd)

Atlantis, In der Henn 27, 5030 Hürth 3, 228 Mark
Softwareland, Franklinstr. 27, CH-8050 Zürich, 229 Mark
Micro-Systems Software Inc., 4301-18 Oak Circle, Boca Raton, Florida 33431

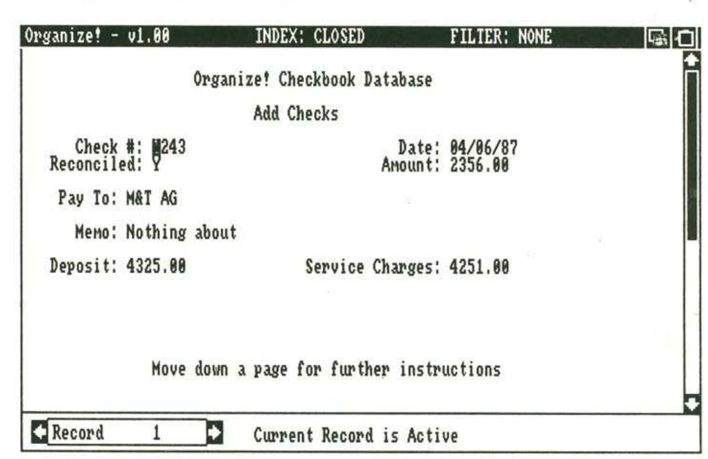
Public Domain in Hülle und Fülle

Software zum Diskettenpreis gibt es seit längerer Zeit auch für den Amiga. Ein Großteil der vorhandenen Public Domain wird von Atlantis angeboten. Zur Verfügung stehen die ersten 58 Fish-Disks, sowie 40 Disketten von Panorama, die allerdings bereits von 1986 stammen.

Daneben werden noch 39 Faug-Disketten, 15 Amicus, 5 SACC und diverse andere Disketten von User-Groups angeboten. Von Atlantis selbst gibt es in der Zwischenzeit 10 eigene Disketten. Zur Auslieferung kommen 2DD-Disketten zum Preis von 6.80 bis 9.80 Mark, je nach Umfang der Bestellung. Gruppenbestellungen lohnen sich also in jedem Fall.

Wer nur mal reinschnuppern will, kann sich gegen eine Schutzgebühr den umfangreichen Katalog von Atlantis schicken lassen. Dort sind die Inhaltsverzeichnisse der einzelnen Disketten mit einer kurzen Inhaltsangabe, allerdings in Englisch, enthalten. (rf)

Atlantis, In der Henn 27, D-5030 Hürth 3, Public Domain, 6.80 bis 9.80 Mark



»Organize!« von Micro-Systems Software Inc.

Die Ideenschmiede

Der Amiga bietet ein weites Betätigungsfeld für die Programmierung. Was dabei herauskommt, wenn sich 18 Top-Programmierer aus aller Welt zusammentun, sieht man am Beispiel »Amilink«. Wir interviewten David Vincent, den General Manager dieser neuen Softwarefirma.

In das mit vier Amigas gut ausgestattete Programmierzimmer der Schweizer Niederlassung von Amilink führte uns der Weg für dieses Interview. Mitten im Herzen von Genf in einer eher unscheinbaren Seitenstraße kocht die Software-Küche auf Hochtouren. Zwischen Hardware, Büchern, Originalprogrammen und natürlich unzähligen Disketten arbeitet eine der interessantesten Entwicklergruppen für Amiga-Software an neuen Programmen, bei denen jeder Amiga-Fan in Verzückung geraten würde.

Amiga-Magazin: Was ist eigentlich Amilink? Aus wieviel Leuten setzt sich die Firma zusammen?

D. Vincent: Amilink ist eine Gruppe von zur Zeit 18 Programmierern, die auf dem Gebiet Amiga-Software an verschiedensten Projekten zusammenarbeiten. Von diesen 18 Leuten sind drei Programmierer aus Deutschland und sechs aus den Staaten. Die neun restlichen sind direkt hier aus der Schweiz.

Amiga-Magazin: Wie ist es zur Gründung gekommen und wie arbeiten diese völlig verschiedenen Menschen zusammen?

D. Vincent: Amilink existiert ungefähr seit September '86 und es sind wirklich nur Programmierer tätig, die sich hervorragend mit der Maschine auskennen und die einfach auf nichts anderem programmieren wollen. Für jeden von uns gilt: »We love this machine«. Das ist die Grundvoraussetzung unter der wir uns zusammengefunden haben. Die Leute, die in der Schweiz sitzen, kommen regelmäßig jeden Monat zu einem Meeting hierher; alle anderen wie zum Beispiel die Programmierer in den Staaten stehen in ständigem telefonischem Kontakt. Wir sind so gut informiert wie keine andere Programmierergruppe sonst, da wir ständig unsere eigenen Programme, Tips, Ideen, Sourcecodes und auch Public-Domain-Programme über DFÜ austauschen.

Amiga-Magazin: Die Programme werden also ausschließlich für den Amiga produziert oder sind auch Umsetzungen auf andere Systeme geplant?

D. Vincent: Zur Zeit wird einzig auf dem Amiga programmiert. Eventuell werden wir einige Spiele, die wir für die nahe Zukunft geplant haben, auch auf anderen 68000-Systemen veröffentlichen. Momentan bringen wir allerdings nur Programme auf den Markt, die wirklich alle speziellen Möglichkeiten des Amiga ausnutzen und daher auf keinen Fall umsetzbar sind.

Amiga-Magazin: Mit welcher Programmiersprache wird bei Amilink am meisten gearbeitet und welche wird sich schätzungsweise im Heimbereich durchsetzen?

D. Vincent: Für den Heimbereich ist dies noch schwer abzuschätzen; es hängt von den Ansprüchen und der jeweiligen Anwendung ab. Wir verwenden im wesentlichen Assembler und C, in einigen Fällen auch Modula 2.

Amiga-Magazin: Wie kann man die Neuentwicklungen Amiga 500 und 2000 einschätzen und wie steht es mit den Marktchancen für die Amiga-Produktlinie?

D. Vincent: Commodore hat den einzig richtigen Schritt getan und eine Produktdifferenzierung beim Amiga vorgenommen. Vorher wußte eigentlich niemand so recht auf welche Anwender der Amiga 1000 genau abzielte. Mit seinen Tendenzen zum PC-Bereich und den tollen Fähigkeiten eines Heimcomputers von morgen ist und bleibt der Amiga eine der interessantesten Maschinen auf dem Markt. Er wird sich natürlich jetzt in seinen zwei Ausführungen viel leichter verkaufen als vorher.

Amiga-Magazin: Wo liegt die Schwierigkeit beim Programmieren mit dem Amiga?

D. Vincent: Was uns immer wieder große Schwierigkeiten bereitet, ist die mangelnde Dokumentation der Systeme, auch im professionellen Bereich. Vieles über den Amiga mußten wir zum Beispiel erst selbst austüfteln, weil wirklich von niemandem Informationen zu bekommen waren. Störend am Amiga selbst ist für uns, die wir sehr viel mit Erweiterungen arbeiten, eigentlich nur, daß er nicht von einem angeschlossenen Festplattenlaufwerk bootet. Aber auch das wird hoffentlich bald behoben sein.

Amiga-Magazin: Was hat sich Amilink für die Zukunft vorgenommen und was für Programme werden wir demnächst sehen?

D. Vincent: Amilink versucht immer Programme zu entwickeln, die etwas wirklich Neues bringen und die die Grenzen des Machbaren auf dem Amiga erweitern. Bereits fertiggestellt ist zum Beispiel ein Malprogramm für den Hold-and-Modify-Modus des Amiga mit seinen 4096 Farben (Bild 1). Dabei hat man dann fantastische Möglichkeiten zum Editieren, die man bisher nur von Zeichenprogrammen in den normalen Auflösungsstufen des Amiga kannte. So gut wie fertig ist inzwischen auch der ultimate Fonteditor für alle Zeichensätze, die man sich nur wünscht. Die einzelnen Zeichen dürfen bis zu 16 verschiedene Farben haben und es kann ein Pattern als Hintergrund entworfen werden. Die Fonts sind dann von jedem Programm, welches auf die Amiga-Fonts auf der Diskette zugreift, verwendbar. Sehr hilfreich für viele Amiga-Anwender wird auch unser Programm zum Erstellen eigener Druckertreiber sein. An Spielen ist zur Zeit erst »Othello-Master« eine Art Reversi-Programm fertiggestellt, was aber mit Spitzengrafik und -komfort aufwartet.

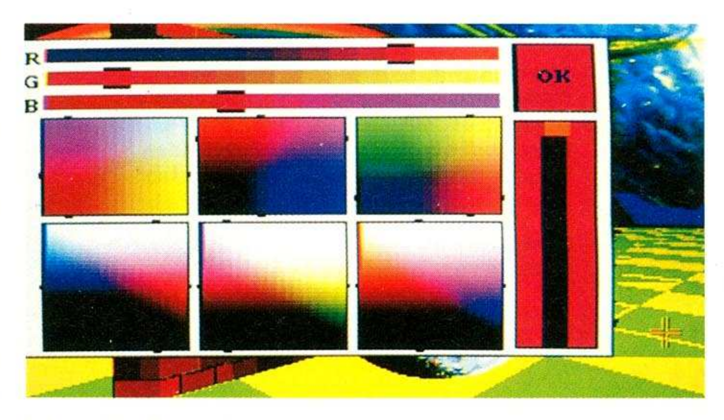


Bild 1. Die Farbpalette von »Prism«, dem neuen Malprogramm für den HAM-Modus

Daß die Amilink-Programmierer in allen Belangen über den Amiga Bescheid wissen, zeigte sich nicht nur an den vielen Programmen, die uns im Laufe der, wie immer viel zu kurzen Zeit, vorgeführt wurden. Verblüffend war auch die geheime Passion von David Vincent, der kleine Gag- und Gimmick-Programme für den Amiga schreibt, um diese vielleicht einmal gesammelt auf einer Diskette herauszubringen. Dabei gibt es dann Routinen wie »Melt«, bei der die Grafik auf dem Bildschirm förmlich schmilzt und nach unten zerläuft, so als wäre es dem Prozessor gerade zu heiß geworden. Oder wenn bei »Tilt« auf einmal der Bildschirm um 45° gekippt wird und wie bei einem Flipperautomaten auch noch als Guru Meditation ein riesiges Tilt-Zeichen erscheint. Diese Gags gehören ebenso zum Repertoire der Programm-Zauberer von Amilink, wie eben mal schnell 150 Fenster auf dem Bildschirm zu öffnen oder eine Workbench mit 16 Farben zu programmieren. Jedenfalls steht uns noch einiges an fantastischer Software für den Amiga ins Haus. (jk)



Wählen Sie den richtigen Einstieg zu Ihrem Amiga 500. Denn das Handbuch läßt Sie dabei völlig im Stich. Versuchen Sie es also lieber gleich mit Amiga 500 für Einsteiger. Hier heißt es: Anschließen und sofort loslegen. Schritt für Schritt finden Sie alle Informationen einsteigergerecht serviert: Die Arbeit mit der Workbench, das erste Programm in AmigaBASIC, das CLI, das AmigaDOS mit den wichtigsten Befehlen – und, was Sie im Handbuch vergeblich suchen werden, eine detaillierte Beschreibung der Zusatzprogramme, die sich auf den mitgelieferten Disketten befinden. Das sind die Informationen, die jeder Einsteiger braucht, um schnell mit seinem neuen Rechner arbeiten zu können.

Amiga 500 für Einsteiger
ca. 300 Seiten, DM 39,–
erscheint ca. 5/87

Das erfolgreiche Buch zu AmigaBASIC – jetzt in der Neuauflage! Erweitert und völlig überarbeitet. Nun auch mit Kickstart 1.2, neuer Workbench und Amiga 500 & 2000. Und allem, was BASIC-Programmierern sonst noch Spaß macht: Grafik und Sound, Laden und Speichern von Graficraft-Bildern in BASIC-Programmme, sequentielle und relative Dateien, Business-Grafik, Computeranimation, Pull-Down-Menüs. Windows, Wellenformen, Umgang mit IFF-Bildern, Sprachausgabe und, und, und. AmigaBASIC für alle. Ob Einsteiger, Aufsteiger oder Profi – in diesem Buch finden Sie alles. Und zwar so, daß Sie es auch verstehen.

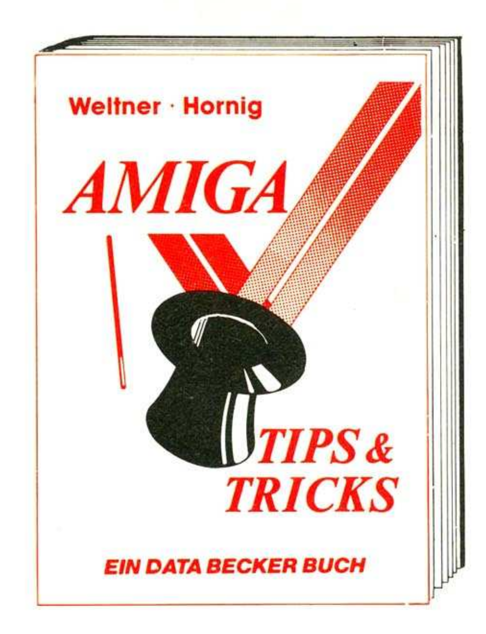
AmigaBASIC
Hardcover, 774 Seiten, DM 59,–

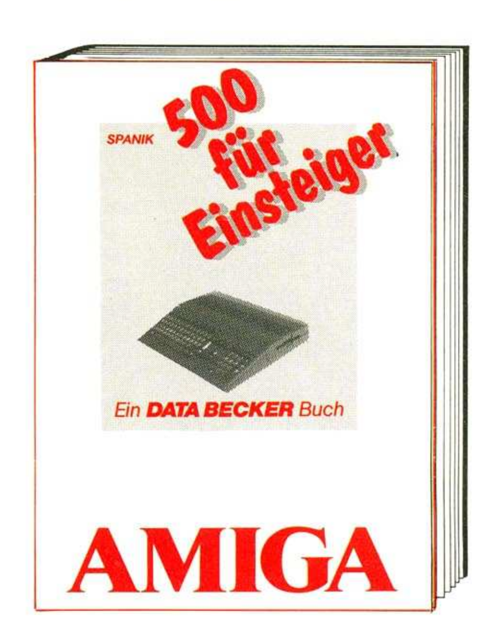
Was leisten die neuen Amigas? In dem Buch „Das können Amiga 500 & 2000" finden Sie die Antwort. DATA WELT Autor und Amiga-Fan Michael Stein hatte die besten Voraussetzungen, es zu schreiben. Lange vor der Markteinführung konnte er direkt mit den Entwicklern die neuen Geräte testen. Hier informiert er Sie nun, was Sie von diesen Rechnern erwarten dürfen. Unabhängig davon, ob Sie den Amiga schon haben oder den Kauf planen: Dieses Buch bietet Ihnen Entscheidungshilfen, technische Details und jede Menge von dem, was man mit Amiga 500 & 2000 so alles anstellen kann. Eben Informationen, die man braucht, wenn man sich für die neuen Amigas interessiert. Aufbereitet nach einem völlig neuartigen, didaktischen Konzept, in einer Sprache, die zum Amiga paßt.

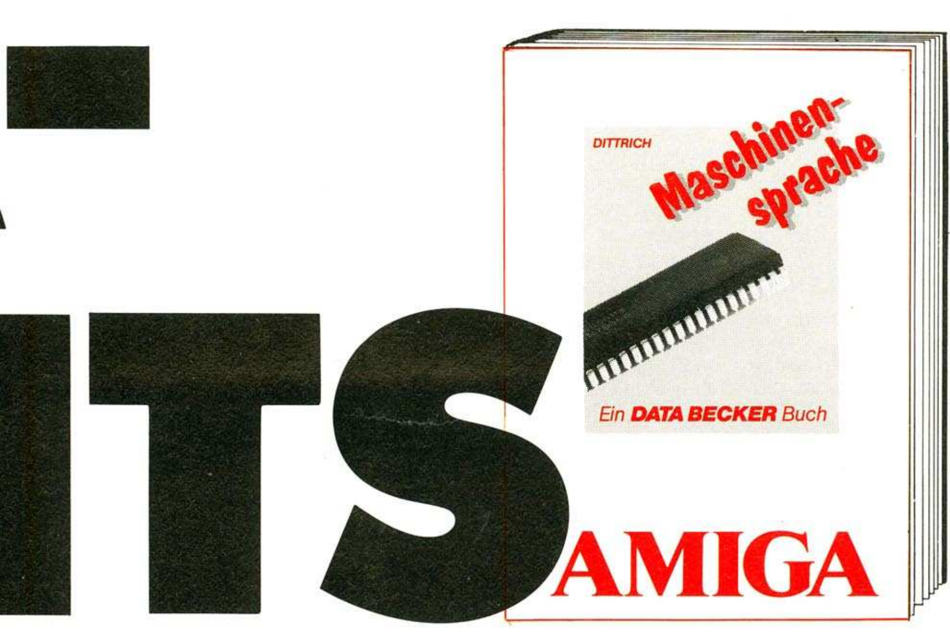
Das können Amiga 500 & 2000
190 Seiten, DM 29,–

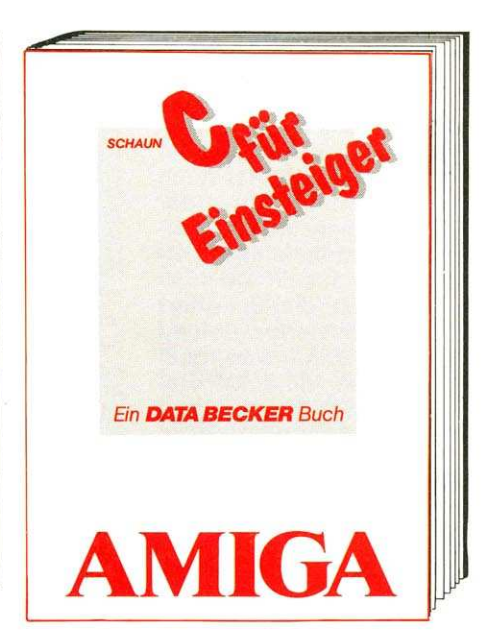
DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010











Schreiben Sie Ihre Programme in Maschinensprache – und Sie werden erst recht sehen, wie schnell der Amiga sein kann. Das nötige Know-how liefert Ihnen das Buch „Amiga Maschinensprache". Hier bekommen Sie genau die Informationen, die Sie brauchen: Speicher-Aufbau des Amiga, Register, Adressierung und Befehlssatz des 68000, das Amiga-Betriebssystem, Druckeransteuerung, Diskettenoperationen, Sprachausgabe ... Aber hier erfahren Sie auch gleich, wie Sie mit den drei wichtigsten vorhandenen Assemblern arbeiten: dem Assembler des Entwicklungspaketes, dem K-SEKA-Assembler und dem PROFIMAT zum Amiga. Amiga Maschinensprache – der Weg zu professionellen Maschinensprache-Programmen.

Amiga Maschinensprache
Hardcover, ca. 300 Seiten, DM 49,–
erscheint ca. 5/87

Die Sprache C an einem Wochenende lernen? Durchaus möglich! Mit C für Einsteiger. Ein Einführungskurs, der Ihnen schnell und einfach die wichtigsten Grundlagen dieser Sprache vermittelt. Vom ersten Programm bis hin zu den Routinen in den Bibliotheken. Variablentypen, Rechnen in C, Schleifen und Bedingungen, Funktionen, Speicherklassen, Ein-/Ausgabe – der gesamte Sprachumfang sowie die besonderen Features von C werden anschaulich und leichtverständlich erklärt. Zahlreiche Tips & Tricks zur Programmierung und eine detaillierte Beschreibung der beiden Compiler Lattice C und Aztek runden das Ganze ab. Schneller und effizienter geht's nicht.

Amiga C für Einsteiger
ca. 300 Seiten, DM 39,–
erscheint ca. 5/87

64 Farben gleichzeitig auf dem Amiga? Von BASIC aus Zugriff auf die Libraries? Erstellen und Benutzung eigener Zeichensätze? Sinnvoller Einsatz von Windows, Screens und Menüs? Verschiedene Fonts unter AmigaBASIC? Wenn das die Dinge sind, die Sie suchen, haben wir genau das Richtige für Sie: Amiga Tips & Tricks. Mit zahllosen Tips zu einzelnen Grafikbefehlen, Programm-und Amiga-DOS-Routinen! Ein Buch, das voller Überraschungen steckt; das Ihnen Anregungen, aber auch fertige Lösungen bietet.

Amiga Tips & Tricks
Hardcover
317 Seiten
DM 49,–

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

DER ZWEITE KLICK

Warum kann man ein zweites CLI-Fenster nur mit dem Befehl »NEWCLI« aufrufen? Es nützt nichts, einfach das CLI-Icon in der System-Schublade nochmals anzuklicken, wenn man dies schon einmal getan hat.
RAINER TASCHKE

DRUCKER-BEFEHLE

Ich besitze einen Matrixdrucker Seikosha SP-1200 AI. Das Drucken mit Programmen oder von Maschinensprache aus ist problemlos möglich. Allerdings habe ich Schwierigkeiten, Sonderbefehle von Basic aus an den Drucker zu senden. Man möchte zum Beispiel auch andere Schriftarten oder Sonderzeichen drucken. Eine Basic-Befehlsfolge wie etwa: PRINT #1, CHR$(27); "M";... um die Schriftart Elite zu aktivieren, zeigen keine Wirkung. Was muß man tun, um alle ESC/P-Befehle von Amiga-Basic aus einsetzen zu können?
CHRISTIAN PROBST

DER TRICK MIT DEN ICONS

Wenn man die Icons auf der Workbench anklickt, verändern sie ihre Farbe. Ich habe inzwischen auf mehreren Utility-Disketten Icons entdeckt, die sich beim Anklicken bewegen oder ein völlig anderes Aussehen annehmen. Wie kann man solche Icons selber konstruieren? **KLAUS MAREI**

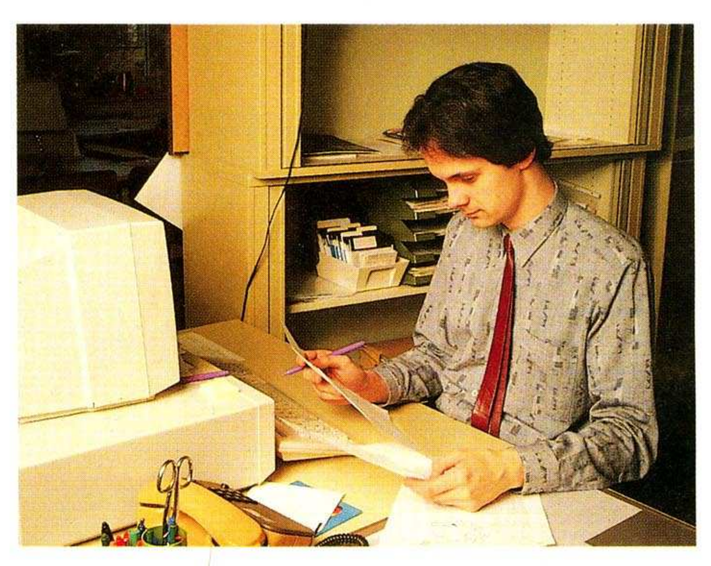
FRAGEN SIE

Wenn Sie Probleme mit dem Amiga oder mit anderen Computern in Zusammenarbeit mit dem Amiga haben, stellen Sie Ihre Fragen ans Leserforum des Amiga-Magazins. So können Sie in Kontakt mit anderen Lesern kommen, die vielleicht dieselben Probleme schon gelöst haben.

DATENTRANSFER

Auf dem Commodore 128 D arbeite ich bereits seit einiger Zeit mit dBase II. Welche Möglichkeiten gibt es, die auf den 5¼-Zoll-Disketten befindlichen Daten für Datenbanken auf dem Amiga verfügbar zu machen?
JOCHEN SCHILLER

Wie kann man Dateien, die mit Superbase 64 auf dem Commodore 64 erstellt wurden, auf den Amiga übertragen, so daß sie in Superbase Amiga weiterverwendet werden können? **P.-W. WAHL**



In der Amiga-Redaktion werden Leserbriefe beantwortet und die interessantesten für das Leserforum ausgewählt

AMIGA-FARBEN

Wie kommt es zur Anzahl von 4096 möglichen Farben, die der Amiga darstellen kann? Was ist eigentlich das Besondere am H.A.M.-Modus und warum werden dabei die Farben an einigen Stellen immer etwas verwischt?
FRANK SCHLENZIG

Für die Farben und Grafikauflösung des Amiga ist der Denise-Chip zuständig. Interessant ist in diesem Fall nicht unbedingt, daß er insgesamt 32 Farbregister besitzt, sondern daß jedes dieser Farbregister 12 Bit breit ist. Für die Grundfarben Rot, Grün und Blau werden dabei jeweils vier Bit zur Einstellung der Intensität gebraucht.

Zur resultierenden Farbe kann also jede Grundfarbe in maximal 16 verschiedenen Abstufungen zugemischt werden. Daraus ergibt sich die insgesamt mögliche Farbanzahl von 16 x 16 x 16 = 4096 Farben. Der H.A.M.-Modus (Hold-and-Modify) des Amiga ist die einzige Methode alle 4096 Farben gleichzeitig auf dem Bildschirm darzustellen.

Der Grafik-Chip Denise ist so gebaut, daß er bis zu fünf Bit-Ebenen im normalen Modus verwalten kann. Die Bit-Ebenen kann man sich als zweidimensionales Feld vorstellen, in dem Nullen und Einsen als Informationsträger für eine Grafik eingetragen sind. Um nun mehrfarbige Grafiken zu erzeugen, werden mehrere dieser Bit-Ebenen sozusagen übereinandergelegt. Aus den entsprechenden Bit-Kombinationen wird ein Punkt auf dem Bildschirm und seine aktuelle Farbe bestimmt. Beim besonderen H.A.M.-Modus wird nun noch eine sechste Bit-Ebene hinzugenommen. In Zusammenarbeit von Bit-Ebene 6 und Ebene 5 werden nun die in den ersten vier Ebenen gesetzten Punkte interpretiert. Dabei wird die Farbe eines Punktes folgendermaßen bestimmt: Der Farbwert des Punktes links vom momentan angesprochenen Punkt wird zunächst einmal ermittelt (Hold). Dann wird, je nachdem welche Bits in Ebene 5 und 6 gesetzt sind, die Intensitätsinformation einer Grundfarbe ersetzt durch die momentan aus Ebene 1 bis 4 resultierende Farbe. Es kann allerdings immer nur der Intensitätswert einer Grundfarbe ersetzt oder modifiziert werden (Modify). Darum braucht man zwischen zwei Farben, die sehr unterschiedliche Farbwerte haben, immer mindestens eine Zwischenstufe. (jk)

FERNSEHMODUL

Da ich zur Zeit zwar das Geld für einen Amiga 500 habe, jedoch nicht für den Monitor 1081, möchte ich einen Farbfernseher anschließen. Aus Ihren Zeitschriften erfuhr ich, daß der Amiga 500 nicht so ohne weiteres an einen Fernseher anschließbar ist und es eventuell ein entsprechendes Modul von Commodore selbst geben wird, womit man dies dann doch bewerkstelligen kann. Ich möchte nun wissen, ob man nicht sonst irgendwo schon ein Modul zur Umwandlung von RGB-Signalen in eine farbfernsehverträgliche Norm beziehen kann?
VOLKER SCHMIDT

AMIGA 500 FRAGEN

1) Wieviel Speicherplatz hat dieser Computer?
2) Programme für den Amiga benötigen oft als Hardware-Anforderung 512 KByte, obwohl sie gar nicht so lang sind und schnell geladen werden. Wie ist das zu erklären?
3) Ich habe gehört, man könne beim Amiga 500 eine Speichererweiterungskarte einstecken. Wie heißt diese Karte? Wo ist sie zu beziehen? Wieviel KByte Speicher hat der Amiga nach dem Einbau?
4) Laufen alle Programme, wenn man den Amiga auf 1 MByte aufrüstet?
ACHIM HUSCHLE

1) Der Amiga 500 hat serienmäßig einen 512-KByte-Speicher.
2) Der relativ hohe Speicherbedarf mancher Programme entsteht nicht etwa durch das Programm selbst, sondern durch Daten, die nachgeladen oder vom Programm erzeugt werden. In erster Linie sind dies zum Beispiel Grafiken, aber auch aufwendige Strukturen wie etwa Zahlen- oder Stringfelder (Arrays) bei der Analyse von Daten brauchen ihren Platz.
3) An der Unterseite des Amiga 500 befindet sich eine Öffnung in die sich eine zusätzliche 512-KByte-Speichererweiterungskarte einstecken läßt. Damit hat man dann 1 MByte zur Verfügung. Diese Karte wird von Commodore selbst herge-

stellt. Es stand zu Redaktionsschluß noch nicht fest, ob der Amiga 500 in Deutschland sofort mit dieser Erweiterung ausgeliefert wird, oder ob man sie nachkaufen muß. Der Preis dieser Karte wird laut Aussagen von Commodore allerdings nicht sehr hoch, mit anderen Worten nicht über dem Preis einer externen Erweiterung, wie sie für den Amiga 1000 erhältlich ist, liegen.
4) Alle Programme, die für und auf dem Amiga 1000 geschrieben wurden, laufen sicherlich nicht auf dem Amiga 500. Dies hat im wesentlichen zwei Gründe: Zum einen ist daran die Kickstart-Version 1.2 schuld, die der Amiga 500 fest im ROM eingebaut hat. Alte Programme, die darauf ausgerichtet sind ausschließlich mit Kickstart 1.1 zu laufen, werden erhebliche Schwierigkeiten verursachen. Des weiteren bezeichnet man alles, was über 512 KByte liegt, beim Amiga als Fast-RAM. Auf dieses RAM können die Coprozessoren des Amiga nicht zugreifen. Wegen der variablen Speicherverwaltung des Amiga können immer einige Programmteile oder Daten in dieses RAM gelegt werden. Dies wird jedoch von älteren Programmen nicht registriert, da diese nicht damit rechnen, daß Speicher oberhalb der 512-KByte-Grenze verfügbar ist. Die Folge ist ein unrettbarer Programmabsturz. Man kann diesem Problem aus dem Weg gehen, indem man das auf der Workbench zum Amiga 500/2000 befindliche Programm »No Fast Mem« aufruft. Dieses sperrt den oberen Speicherbereich, und man hat nur 512 KByte zur Verfügung. Schwierig ist nur, daß man das Laden und Aufrufen dieses kleinen Programms oftmals in die Startup-Sequence vieler Original-Programme einbinden müßte, da diese nur nach einem Reset hochfahren. Dazu muß sich »No Fast Mem« natürlich auch auf der jeweiligen Diskette befinden. Zusammenfassend kann man sagen, daß die Software, die in Zukunft herauskommt, diese Probleme nicht kennen wird, da sie darauf ausgerichtet ist, auch auf Amigas mit großem Speicher zu laufen (bekanntlich bis 8 MByte). Ältere Programme werden, soweit sie gut genug sind, sicherlich von den Herstellern auf einen größeren Speicher angepaßt werden, oder zumindest werden Update-Versionen erstellt, die die oben beschriebenen Probleme nicht kennen. (jk)

AMIGA 1000 KARTEN

1) Welche Schnittstellen entsprechen beim Amiga 1000 nicht dem Standard?
2) Bleiben künftige Erweiterungen weiterhin auch für den Amiga 1000 verwendbar?
3) Kann man die Steckkarten des Amiga 2000 auch für das Sidecar verwenden?
4) Ist auf den Erweiterungskarten für den Amiga 1000 eine batteriegepufferte Echtzeituhr oder ist diese nur auf der 512-KByte-Karte des Amiga 500 angebracht? Kann man vielleicht diese Amiga 500-Karte am Amiga 1000 verwenden?

MICHAEL GRIEGER

1) Die bidirektionale Schnittstelle zur seriellen Datenübertragung, kurz RS232-Schnittstelle, sowie auch die Centronics-Schnittstelle des Amiga 1000 entsprechen nicht ganz dem üblichen PC-Standard. Gefährlich kann dabei vor allem die an Pin 23 der Centronics-Schnittstelle anliegende 5-Volt-Spannung sein, die, sofern sie falsch verbunden wird, zu ernsthaften Hardware-Schäden am Amiga 1000 oder angeschlossener Peripherie führen kann. Der Amiga 500 hat diese Probleme nicht, da er über genormte Schnittstellen verfügt.
2) Die Erweiterungen, die an den extern herausgeführten Bus des Amiga angeschlossen werden, sind grundsätzlich auf allen Modellen verwendbar. Es kann nur zu Schwierigkeiten aufgrund unterschiedlicher Kickstart-Versionen oder wegen der hardwaremäßig verschiedenen Anordnung des Expansion-Slots kommen.
3) Inwieweit sich die Steckkarten des Amiga 2000 und des Sidecar vertragen beziehungsweise austauschen lassen, werden wir ausführlich in einer der nächsten Ausgaben untersuchen; sobald der Amiga 2000 inklusive Steckkarten verfügbar ist.
4) Einige der zur Zeit auf dem Markt erhältlichen Erweiterungskarten für den Amiga 1000 verwenden eine batteriegepufferte Echtzeituhr. Auf der internen 512-KByte-Erweiterungskarte des Amiga 500 befindet sich auf jeden Fall eine solche Uhr. Die Karte für den Amiga 500 kann man auf keinen Fall im Amiga 1000 verwenden, da sie darauf ausgelegt ist, intern im Computer eingesteckt zu werden. Dies geschieht an einer Steckleiste, über die nur der Amiga 500 verfügt und die nicht identisch mit dem Expansions-Slot aller Amiga-Modelle ist. (jk)

LUST UND FRUST AUF DEM AMIGA

Seit Anfang des Jahres sind mein Mann und ich stolze Besitzer eines Amiga 1000. Gekauft haben wir diesen »Supercomputer« obwohl oder gerade weil wir absolute Computer-Neulinge waren. Mit mehr Ahnung wären uns beim Kauf sicher eine Menge Fragen eingefallen, auf die wir erst nach und nach stoßen. Unbeleckt von jeder Sachkenntnis haben wir diese erste Hürde genommen. Brav sind wir dann beim Aufstellen und Starten der Anleitung gefolgt und haben auch gleich die ersten Kopien von den mitgelieferten Disketten gemacht. Dann wollten wir natürlich ausprobieren, was es so alles gab. Hinter uns standen zwei sehr ungeduldige Kinder, deren Meinung nach ein Computer natürlich zum Spielen da ist. Und damit ging's los: Können Sie sich vorstellen, wie lange ein absoluter Anfänger braucht, bis er merkt, daß er keinen Kopierfehler gemacht hat, sondern daß die »Mindwalker«-Diskette und auch die Musik-Demos einen nirgends erwähnten Kopierschutz haben?

Inzwischen sind wir mehr oder weniger um diese und eine Reihe anderer Hindernisse herumgekurvt; haben uns immer wieder gefreut, wenn wir Probleme lösen konnten, die in den Handbüchern nicht auftauchen. Man lernt seinen Amiga jeden Tag besser kennen. Auch das Fachchinesisch wird nach und nach immer klarer. Allerdings bin ich nicht mit dem Urteil vieler Fachjournalisten über die Qualität der Handbücher einverstanden. Wer sich bereits einigermaßen mit Computern auskennt, mag alles finden, was er braucht. Bei uns fingen die ersten Probleme schon mit der Tastatur an, von der es nicht einmal eine genaue Abbildung gibt. Dann fehlt ein Sachwortregister, eine übersichtliche und leicht zu findende Befehlszusammenstellung, eine Gegenüberstellung verschiedener Möglichkeiten, ein und dasselbe zu erreichen, eine vollständige Auflistung aller Fehlermeldungen, egal wo sie im DOS oder im Basic auftauchen, und so weiter. Eine sehr gute Quelle für absolut notwendige und nützliche Hinweise sind neben Büchern und Zeitschriften meiner Meinung nach die Public Domain-Disketten. Obwohl ich im Englischen nicht sehr fit bin, habe ich gerade dort sehr viel gelernt.

Weil wir aber der Meinung sind, daß selbst der schönste Computer ohne Software nichts nutzt, haben wir die einzig zur Verfügung stehende damals gleich mitgekauft: das Anwenderpaket von Commodore mit Graficraft, Textcraft, dem MS-DOS-Transformer und dem M&T-Amiga-Handbuch. Abgesehen davon, daß zu Textcraft nur eine Kurzreferenz beilag, stimmt diese noch nicht einmal mit der Version Plus D überein. Zwar kommt man auch so ganz gut zurecht, aber ich suche immer noch verzweifelt nach der sehr nützlich erscheinenden Anzeige einer ganzen Seite, was im Buch beschrieben ist. Mit MS-DOS habe ich auch noch ungelöste Probleme. Ich hoffe dazu fehlende Tips und Tricks noch irgendwo zu finden.

Warum ich noch längst nicht alles in eine Ecke geschmissen habe? Es macht natürlich trotzdem noch riesigen Spaß. Selbstverständlich gibt es auch hilfsbereite Fachleute, die auch die dümmsten Fragen beantworten und bei schwierigen Problemen sogar mit nach der Lösung suchen. Diesen Mitmenschen sei hier noch einmal ausdrücklich gedankt. All die Dinge, die wir mit viel Mühe und Schweiß herausbekommen haben, sitzen sicher viel besser im Gedächtnis, als wenn wir sie einfach in der Anleitung nachgelesen hätten. Oder? HEIDRUN WOLF

ANTWORTEN SIE

Haben Sie schon eine Lösung zu bestimmten allgemeinen Problemen gefunden, die viele Leser interessiert, dann schreiben Sie uns ihre Antwort oder Lösungsvorschläge. Durch ein Plus an Informationen kann die Gemeinde der Amiga-Freaks schneller wachsen.



Das Amiga Handbuch

Eines der ersten Bücher, das als Handbuch für den Amiga auf den Markt kam, ist »Das Amiga Handbuch«. Es erschien bereits Mitte 1986 und gehörte damals eine ganze Weile lang zur Grundausstattung des Amiga. Eigentlich ist es schade, daß Commodore dies nicht weitergeführt hat, denn man findet in dem Buch viele nützliche Erklärungen und Tips, die für den Einsteiger von Bedeutung sind.

Schon im ersten Kapitel wird es interessant, denn hier beschreibt der Autor die hervorragenden Fähigkeiten dieses Computers. Richtig interessant wird es dann im folgenden Abschnitt des Buches. Hier wird das Arbeiten mit der Workbench genauestens beschrieben und auf alle Eigenheiten dieser Benutzeroberfläche eingegangen. Dabei sollte man während der Lektüre des Buches gleich alles, was man erfährt, auf dem Computer ausprobieren, um so das eben Gelernte anzuwenden. Um »Objekte der Workbench« sowie um »Werkzeuge und Projekte« geht es dann im nächsten Kapitel. Zu diesen gehören beispielsweise alle Arten von Icons (Symbolen) und die Programme, die fester Bestandteil der Workbench sind. Besondere Beachtung findet in diesem Zusammenhang die Aufgabe von Schubladen und die Arbeit mit ihnen.

Was bedeutet der Begriff »Intuition«? Diese und ähnliche Fragen beantwortet der Autor im darauffolgenden Kapitel. Intuition ist, grob gesagt, das Betriebssystem des Amiga. Es ist zum Beispiel für die Maus-Steuerung und für die Verwaltung von Fenstern und Screens verantwortlich. Auch ein Kapitel über das »Arbeiten mit dem Amiga« fehlt nicht. Dort wird auf die Benutzung von damals schon erhältlichen Programmen wie Graphicraft, Deluxe-Paint, Textcraft und Musicraft eingegangen.

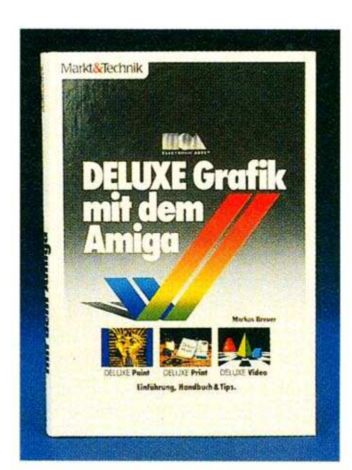
Auch die »Grundlagen des CLI« kommen nicht zu kurz, ihnen ist ein eigener Teil des Buches gewidmet. In direktem Zusammenhang dazu stehen auch die Dateien und Dateiverzeichnisse und deren Gebrauch vom CLI aus. Außerdem werden sämtliche Kommandos des CLI beschrieben, wobei der Text-Editor »ed« ein eigenes Kapitel füllt. Nachdem dieser Teil des Buches mit einigen wichtigen Tips für das CLI abgeschlossen wurde, folgen Kapitel über die Chips, Grafik, Sound, und die Schnittstellen des Amiga. Abschließend finden sich Hinweise über die Programmierung aller Sprachen, speziell wird dabei auf Basic eingegangen. Dazu sei aber bemerkt, daß die letzten Kapitel nicht mehr sehr umfangreich sind und mehr zur Abrundung des 450 Seiten starken Werks dienen. Das Buch kann allen Anfängern empfohlen werden, die einen Begleiter auf der Reise in die Welt des Amiga nicht missen wollen. (M.Marek/rb)

Markus Breuer, Das Amiga Handbuch, Markt & Technik Verlag AG, 461 Seiten, ISBN 3-89090-228-6, Preis 49 Mark

AMIGA Programmier-Praxis mit MS BASIC

In jedem Computerbesitzer erwacht früher oder später einmal der Wunsch, seinen Computer mit Hilfe einer Programmiersprache nach seinen Bedürfnissen zu steuern. Dabei fällt die Wahl meist auf Basic, eine Sprache, die praktisch zu jedem Computer in verschiedenen Dialekten erhältlich ist. Zum Erlernen dieser Sprache, die auf dem Amiga aufgrund der zahlreich vorhandenen Kommandos besonders leistungsfähig ist, können Bücher gute Dienste leisten. Im Tewi-Verlag ist zu diesem Thema ein Buch mit dem Titel »AMIGA Programmier-Praxis mit MS BASIC« erschienen. Das ursprünglich von dem Amerikaner David A. Lien geschriebene Werk wurde gut ins Deutsche übersetzt. Es ist für den blutigen Anfänger auf dem Amiga gedacht, denn der Autor beschreibt alle Befehle und Funktionen, die es in Amiga-Basic gibt, wobei der Leser behutsam von leichten zu schwierigeren Aufgaben geführt wird. Von Kapitel zu Kapitel kommen neue Befehle hinzu, die anhand von kurzen Programmbeispielen erklärt werden. Diese können dann durch Aufgabenstellungen am Ende jedes Kapitels geübt werden. Die Lösungen zu den Problemstellungen können hinten im Buch nachgeschlagen werden. Da es im ersten Teil um die elementaren Anweisungen des Microsoft-Basic geht, ist es für diejenigen, die schon mit einem Heimcomputer in Basic programmiert haben ein leichtes, da einzusteigen, wo es für sie interessant wird. Der Lerneffekt des Buches ist sicher nicht zu unterschätzen, aber leider kommen die Animationsbefehle etwas kurz. An manchen Stellen wird zu wenig auf alle Möglichkeiten bestimmter Befehle eingegangen, aber durch das vermittelte Wissen sollte man sich zum Experimentieren inspirieren lassen. Fest steht, daß man nach intensivem Selbststudium mit diesem Buch das MS-Basic beherrscht. Der Schreibstil des Buchs ist locker und bestimmt nicht zu trocken. Alles in allem ist das Werk jedem zu empfehlen, der mit dem Amiga-Basic, das er schließlich mitgeliefert bekommt, etwas Konkretes anfangen will. (M. Marek/rb)

David A. Lien, AMIGA Programmier-Praxis mit MS BASIC, Tewi-Verlag, 400 Seiten, ISBN 3-92180369-1, Preis 59 Mark



DELUXE Grafik mit dem Amiga

Wer einen Amiga besitzt und sich kreativ mit diesem für Grafik und Soundanwendungen prädestinierten Computer beschäftigen will, wird an den hervorragenden Produkten der Deluxe-Serie von Elektronic Arts kaum vorbeikommen. Vor allem Deluxe-Paint läßt die Herzen von Grafikfans höher schlagen. Leider sind jedoch die zu diesen Programmen ausgelieferten Handbücher meist nicht ausführlich genug, was die Entstehung eines Buches zur Folge hatte, das sich speziell mit den Programmen Deluxe-Paint, Deluxe-Print und Deluxe-Video auseinandersetzt.

Auf knapp 370 Seiten beschreibt »DELUXE Grafik mit dem Amiga« alle Feinheiten dieser drei Produkte, die man kennen sollte, wenn man die Programme besitzt und mit den Handbüchern nicht zufrieden ist. Mit dem Kauf des Buches erspart man es sich sogar, die mitgelieferten Bedienungsanleiungen zu studieren, denn in diesem Buch findet sich alles besser beschrieben und ergänzt durch zahlreiche andere Informationen.

Zu Beginn findet sich Grundsätzliches zur Grafikerstellung und zu den visuellen Möglichkeiten des Amiga. Für den Neuling ist auch noch die Arbeit mit der Workbench und dem CLI kurz beschrieben. Danach geht es zur Sache mit dem großen Kapitel über Deluxe-Paint. Auch hier wird zuerst das elementare Wissen über Zweck und Fähigkeiten dieses Programms vermittelt, um anschließend sofort mit der Arbeit beginnen zu können. Alle Funktionen von Deluxe-Paint sind dort ausführlich beschrieben.

Weiter geht es dann mit dem Kapitel, das sich mit Deluxe-Print auseinandersetzt. Dieser Teil des Buches nimmt den wenigsten Platz in Anspruch, da Deluxe-Print nicht so viele Funktionen besitzt wie zum Beispiel Deluxe-Video, mit dem sich der nächste Teil des Buches beschäftigt. Gerade bei dieser Art von Software, die dazu dient, elektronische Zeichentrickfilme zu zeichnen, kann man vielerlei Tips und Arbeitserleichterungen geben.

Den Abschluß des Werks bildet ein Kapitel über weitere Hardware, die man an den Amiga anschließen kann. Hier werden dem Leser Begriffe wie Genlock, Festplatten sowie Grafik- und Sounddigitizer erklärt.

Insgesamt gesehen kann das Buch dem Anwender, der die drei Programme wirklich ausnützen will, durchaus empfohlen werden. (M. Marek/rb)

Markus Breuer, DELUXE Grafik mit dem Amiga, Markt & Technik Verlag AG, 370 Seiten, ISBN 3-89090-412-2, Preis 49 Mark

Der KAUFHOF bietet Datensicherheit

BASF Spitzentechnologie – jetzt im Komplettangebot incl. BASF FlexyDisk® Archivbox

Soll man sich bei Disketten, dem wichtigsten Bindeglied zwischen Programm und Computer, auf Ware zweifelhafter Qualität verlassen? Kleinste Datenausfälle, „drop outs", können die Arbeit von Tagen zunichte machen. Daher ist absolute Sicherheit ein wichtiger Maßstab für die Qualität einer Diskette. Wer BASF zu seinem Partner macht, wählt damit gleichzeitig die Datensicherheit in hoher Perfektion. Dafür sorgen umfangreiche Qualitätstests in allen Fertigungsstufen und die Wahl nur bester Materialien.

Wir bieten Datensicherheit zum Sparpreis, BASF® FlexyDisk als Komplettangebot in der Kunststoff-Archivbox. Ob 5.25" oder 3.5", ob einseitig oder zweiseitig, wir bieten für alle Computersysteme die passende Diskette.



Bestellschein

bitte einsenden an: Kaufhof AG, KE 614-Bürotechnik, Postfach 10 10 08, 5000 Köln 1

Ich bestelle:

..... 10er Pack BASF FlexyDisk 5.25"/1 D in Archivbox zu je DM 24,95

..... 10er Pack BASF FlexyDisk 5.25"/2 D in Archivbox zu je DM 34,95

..... 10er Pack BASF FlexyDisk 3.5"/1 D in Archivbox zu je DM 49,95

..... 10er Pack BASF FlexyDisk 3.5"/2 D in Archivbox zu je DM 69,95

Name/Vorname: ______________________

Straße/Hausnummer: ______________________

PLZ/Ort: ______________________

Auf Postkarte kleben, mit 60 Pf. frankieren, Brief: 80 Pf. Porto!

® = eingetragenes Warenzeichen der BASF-Aktiengesellschaft

Amiga-Basic — Ein Riese mit Schwächeanfällen?

Zum Lieferumfang des Amiga gehört eine Diskette, die das Amiga-Basic enthält. Wir zeigen Ihnen im folgenden Bericht die Stärken und Schwächen dieser modernen Programmiersprache.

Der von Commodore zum Amiga mitgelieferte Basic-Interpreter besitzt sehr leistungsstarke Befehle, die auf die vorhandene Hardware zugeschnitten sind. Getestet haben wir das zum Amiga 2000 mitgelieferte Basic in der Version 1.2, das von Microsoft entwickelt wurde. Dies hat den Vorteil, daß sich auf MS-DOS-Computern in GW-Basic geschriebene Programme leicht auf den Amiga übertragen lassen.

Amiga-Basic befindet sich auf der »Extras«-Diskette und kann durch einfaches Anklicken mit der Maus geladen werden. Es genügt aber auch, ein vorhandenes Basic-Programm mit der Maus anzuwählen: Dieses Programm wird zusammen mit dem Basic geladen und dann automatisch gestartet.

Nach dem Laden öffnet das Amiga-Basic zwei Fenster, die wie üblich mit der Maus aktiviert werden können. Das linke, größere Fenster ist das Direktmodus-Fenster, das heißt alle Befehle, die Sie hier eingeben, werden sofort ausgeführt. Das rechte Fenster ist das List-Fenster; hier schreiben Sie Ihre Programme (Bild 1). Diese Aufteilung in zwei Fenster ist ein entscheidender Vorteil von Amiga-Basic gegenüber anderen Varianten: Auf Zeilennummern können Sie in Zukunft getrost verzichten. Sprungziele erreichen Sie statt dessen über alphanumerische und numerische Marken, die Sie sich nach eigenem Geschmack definieren können.

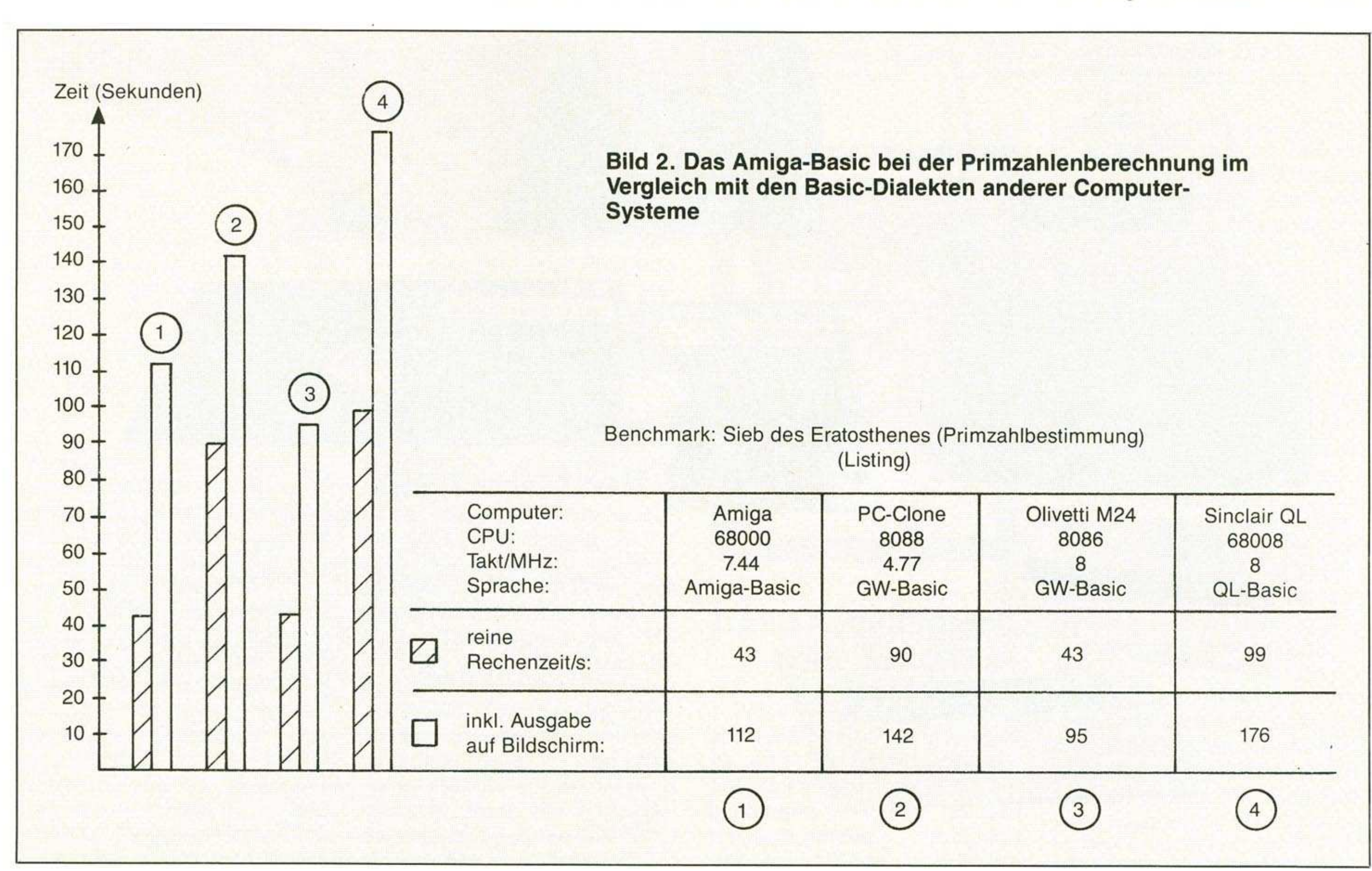
Struktur ist Trumpf

»Wagt« man, ein nach dem ersten Editieren normalerweise nicht fehlerfreies Programm laufen zu lassen, wird eine eventuell falsch eingegebene Zeile im List-Fenster orange umrahmt angezeigt. Hat man sich jedoch nicht verschrieben, aber das Programm produziert dennoch falsche Ergebnisse, dann hilft vielleicht der Trace- und Single-Step-Modus weiter, der es einem erlaubt, Programme schrittweise abzuarbeiten und so Fehler schneller zu finden.

Das Amiga-Basic stellt dem Anwender eine ganze Reihe von strukturierten Steuerbefehlen zur Verfügung, die die Übersichtlichkeit eines Programms erheblich erhöhen. Neben den normalen FOR-NEXT-Schleifen gibt es auch noch IF-THEN-ELSE, END-IF und WHILE-WEND-Strukturen. Wertvolle Hilfen bieten dazu ON ERROR GOTO, um Fehler abzufangen, ON BREAK GOSUB, um bei Programmunterbrechungen ein Unterprogramm ausführen zu lassen sowie die Möglichkeit mit ON TIMER(n) GOSUB alle n Sekunden ein Unterprogramm aufzurufen.

Sowohl für die Abfrage der linken (Auswahltaste), als auch der rechten Maustaste (Menütaste) hält Amiga-Basic Funktionen bereit. Die MOUSE-Befehle beziehen sich dabei auf die Auswahltaste, während die MENU-Befehle auf die Menütaste bezogen sind. Durch diese Instruktionen wird es möglich, die Benutzeroberfläche von Basic-Programmen in der von »Intuition« bekannten Form zu gestalten und damit dem Programm einen professionellen Rahmen zu geben. Ein Beispiel für die Programmierung von Menüs ist in Listing 1 zu finden.

Amiga-Basic trennt, genau wie das Amiga-DOS, nicht scharf zwischen Dateien und Peripheriegeräten. Dies hat den erheblichen Vorteil, daß der Zugriff auf beide Arten von



	Amiga	PC-Clone	Olivetti M24	Sinclair QL
Computer: CPU: Takt/MHz: Sprache:	Amiga 68000 7.44 Amiga-Basic	PC-Clone 8088 4.77 GW-Basic	Olivetti M24 8086 8 GW-Basic	Sinclair QL 68008 8 QL-Basic
reine Rechenzeit/s:	43	90	43	99
inkl. Ausgabe auf Bildschirm:	112	142	95	176
	1	2	3	4

Bild 2. Das Amiga-Basic bei der Primzahlenberechnung im Vergleich mit den Basic-Dialekten anderer Computer-Systeme

Ein-, beziehungsweise Ausgabemedien mit der gleichen Syntax erfolgen kann. Voreingestellt verfügt das Amiga-Basic über die Gerätebezeichnungen DF0, DF1, DF2, DF3 und RAM für die (virtuellen) Diskettenlaufwerke, SER und COM1 für die serielle Schnittstelle und PAR, PRT und LPT1 für die parallele Schnittstelle. An dieser Aufzählung fallen einige Besonderheiten auf: der seriellen Schnittstelle sind zwei Namen zugewiesen, den parallelen sogar drei. Der Grund dafür ist der, daß Amiga-Basic diese Schnittstellen unterschiedlich verwaltet. So können über die PRT- beziehungsweise LPT1-Schnittstelle nur die Steuerzeichen an einen Drucker übergeben werden, die mit dem »Preferences«-Menü eingestellt wurden — alle anderen Steuerzeichen werden herausgefiltert. Die Ursache für die beiden Namen PRT und LPT1 liegt in Kompatibilitätsgründen zu anderen Basic-Dialekten: Microsoft hat in seinen Basic-Entwicklungen (unter anderem GW-Basic auf MS-DOS-Computern) den Namen LPT1 für den Drucker definiert. Der einzige kleine Unterschied zwischen PRT und LPT1 ist, daß LPT1 sofort ausdruckt, während PRT einen Puffer anlegt. Der Unterschied zwischen SER und COM1 ist ähnlich: SER ist eine »normale« Schnittstelle; COM1 jedoch erlaubt die Angabe von Übertragungsparametern (Baudrate, Parität, etc.). Diese Eigenschaft von COM1 vereinfacht die Kommunikation mit anderen Computern über ein Modem oder einen Akustikkoppler.

Zum Anlegen von Dateien unter Basic gibt es zwei Möglichkeiten: einerseits das Schreiben in, Anhängen an und Einlesen von sequentiellen Dateien; andererseits das Arbeiten mit relativen Dateien. Nach dem Öffnen der Dateien mit der jeweiligen Option (»O« für sequentielle Ausgabe, »A« für sequentielles Anhängen, »I« für sequentielles Einlesen und »R« für relative Dateien) kann auf die sequentiellen Dateien mit WRITE#, PRINT# und INPUT# zugegriffen werden. Das Äquivalent für relative Dateien sind die Befehle GET#, PUT#, RSET und LSET. Auf diese Zugriffsmethoden jetzt spezieller einzugehen, würde allerdings viel zu weit führen.

Zu den Dateieingabe- und Ausgabebefehlen gehören natürlich auch noch die Instruktionen zum Laden und Speichern von Programmen. Selbstverständlich ist es von Amiga-Basic aus möglich, nicht nur auf Dateien im aktuell eingestellten Inhaltsverzeichnis zuzugreifen, sondern auch auf alle anderen. Dazu ist entweder der vollständige Pfadname anzugeben oder aber das gewünschte Directory mit dem Befehl CHDIR einzustellen. Um sich über die im jeweiligen Inhaltsverzeichnis vorhandenen Files zu informieren, genügt es den Befehl FILES einzugeben. Es werden dann alle Dateinamen im Ausgabefenster angezeigt.

Wie Sie bei Ausführung von FILES feststellen, befinden sich für jeweils ein Basic-Programm zwei Files im Inhaltsverzeichnis. Eines mit dem Programmnamen und eines, bei dem an den Programmnamen noch ».info« angehängt ist. Dieses ».info«-File enthält Informationen über das Basic-Programm und unter anderem das »Icon«, eine kleine Grafik, mit dem das Programm unter der Benutzeroberfläche »Intuition« angezeigt wird. Während in der alten Amiga-Basic-Version (1.1) die Befehle zum Löschen (KILL) und Umbenennen (NAME) auf jede der beiden Dateien separat angewandt werden mußten, genügt jetzt zum Beispiel KILL <Filename>, um beide Dateien zu löschen.

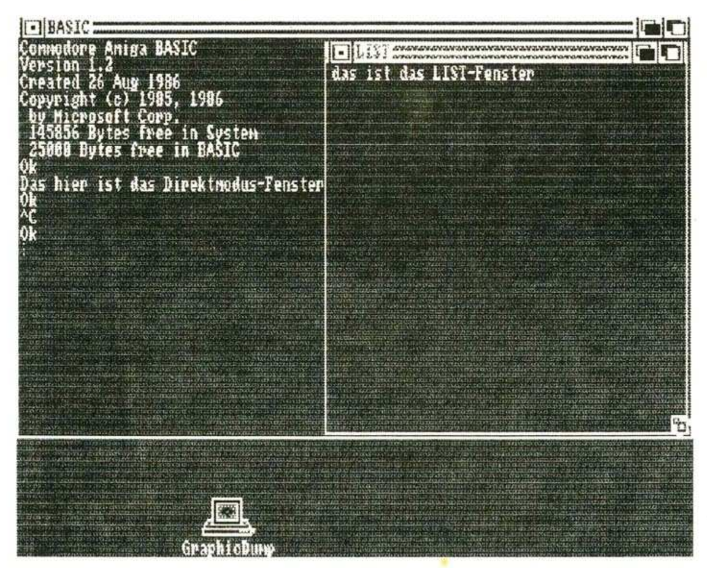


Bild 1. Die Bildschirmaufteilung in Amiga-Basic

```
REM Menüsteuerung - sehr einfach
MENU 1,0,1,"Menü 1"
MENU 2,0,1,"Menü 2"
FOR i = 1 TO 3
  MENU 1,i,1,"Punkt 1."+CHR$(48+i)
  MENU 2,i,1,"Punkt 2."+CHR$(48+I)
NEXT i
MENU 1,4,1,"Beenden"
ende = 0
ON MENU GOSUB Wahl
MENU ON
WHILE ende = 0 : WEND
MENU RESET
END

Wahl:
  m% = MENU(0) : p% = MENU(1)
  LOCATE 10,1
  PRINT "Menü  "+CHR$(48+m%)
  PRINT "Punkt "+CHR$(48+p%)
  IF m% = 1 AND p% = 4 THEN ende = 1
RETURN
```

Listing 1. Eine kleine Menüsteuerung

```
REM Sieb des Eratosthenes bis n = 5000
zeit = TIMER
DIM prime%(5000)
FOR i = 2 TO 2500
  IF prime%(i) = 0 THEN
    j = i + i
    WHILE j <= 5000
      prime%(j) = 1
      j = j + i
    WEND
  END IF
NEXT i
rechenzeit = TIMER - zeit
FOR i = 2 TO 5000
  IF prime%(i) = 0 THEN print i
NEXT i
zeit = TIMER - zeit
PRINT rechenzeit,zeit
END
```

Listing 2. Das »Sieb des Eratosthenes«

Datenstrukturen

In Amiga-Basic gibt es fünf verschiedene Variablentypen: ganze Zahlen (Integer 16 Bit), doppelt genaue ganze Zahlen (32 Bit), Realzahlen (32 Bit, 7 Nachkommastellen), doppelt genaue Realzahlen (64 Bit, 15 Nachkommastellen) und Stringvariablen (Länge bis zu 32767 Byte). Es ist vielleicht noch ganz nützlich zu wissen, daß das Basic die Schleifenvariablen bei FOR-NEXT intern immer als einfach genaue Realzahlen behandelt. Wählt man zum Beispiel im Sieb des Eratosthenes (Siehe Bild 2 und Listing 2) die Schleifenvariablen explizit als Integer, so wird die Rechenzeit um einen Faktor von ungefähr 2,5 verlängert. Zur Manipulation von Variablen existieren sehr viele numerische und alphanumerische Funktionen. Vor allem die Funktionen zur Behandlung von Zeichenketten sind derart umfangreich, daß wohl kaum noch Wünsche offenbleiben.

```
REM Hang Up
CLS : CLEAR
DIM hull%(255)
FOR i = 0 TO 255
  hull%(i) = INT(255*RND - 128)
NEXT i
WAVE 0,hull% : WAVE 1,hull%
WAVE 2,hull% : WAVE 3,hull%
PSET(10,90 - 3*hull%(0)/4)
FOR i = 1  TO 255
  LINE -(10+124/54*i,90-3*hull%(i)/4)
NEXT i
FOR i% = 440 TO 880 STEP 5
  LOCATE 1,1 : PRINT i% : LOCATE 1,10
  PRINT FRE(-2),FRE(-1),FRE(0)
  SOUND WAIT
  SOUND i%,5,150,1
  SOUND i%/2,5,175,0
  SOUND i%*2,5,125,2
  SOUND i%*4,5,100,3
  SOUND RESUME
NEXT i%
END
```

Listing 3. Ein kleines Sound- und Grafikdemo

```
REM Kugel in Auflösung 320 * 400
REM in 32 Farben
SCREEN 1,320,400,5,3
WINDOW 2,"Kugel",,31,3
WINDOW OUTPUT 2
PALETTE 0,.2,.3,1 : REM Hintergrund
FOR i = 1 TO 31
  col = i/31
  PALETTE i,1,col,col : REM Farben
  x = 150 + 1.5 * i
  y = 200 - 2   * i
  r = 180 - 5.5 * i
  CIRCLE (x,y),r,i,,,1.75
  PAINT (x,y),i
NEXT i
END
```

Listing 4. Das Zeichnen einer Kugel in Amiga-Basic

Bei den numerischen Funktionen dürften wohl nur Rechenfanatiker einige transzendente Funktionen (wie Arkuscosinus, Arkussinus, etc.) vermissen. Die numerischen Funktionen können selbstverständlich von allen numerischen Variablentypen benutzt werden.

Schall und Rauch

Das in Listing 3 wiedergegebene Programm sollte eigentlich ein einfaches Beispiel für die Möglichkeiten der Sound-Programmierung werden, aber... davon später. Amiga-Basic stellt zur Erzeugung von Tönen zwei Befehlsarten zur Verfügung. Zum einen die Befehle TRANSLATE und SAY; mit ihnen ist es möglich einen (englischsprachigen) Text in eine Lautschrift zu übersetzen (TRANSLATE) und diese dann vom Computer aussprechen zu lassen (SAY). Die Ergebnisse sind zwar nicht unbedingt berauschend, aber zumindest verständlich. Da das Programm eine amerikanische Entwicklung ist, ist die Aussprache englisch, das heißt deutsche Sätze haben einen amerikanischen Akzent. Mit etwas Mühe kann man jedoch selbst die Lautschrift der deutschen Sätze bestimmen und diese dann an SAY übergeben — dies führt zu etwas besseren Ergebnissen.

Die andere Möglichkeit zur Erzeugung von Tönen gestattet eine weitaus kreativere Gestaltung als SAY und TRANSLATE. Mit ihr kann ein vierstimmiger Tongenerator Frequenzen zwischen 20 und 15000 Hertz (Schwingungen pro Sekunde) erzeugen, wobei sich die Wellenform frei gestalten läßt. Die Tonkanäle 0 und 3 sprechen den linken, die Kanäle 1 und 2 den rechten Tonausgang des Amiga an. Die Frequenz, Lautstärke, Dauer und Kanalnummer des Tones gibt man einfach nach dem Befehl SOUND an. Die Form der Welle läßt sich ebenso einfach mit der WAVE-Anweisung festlegen. Nach dem Kanal nennt man entweder das Schlüsselwort SIN für eine sinusförmige Wellenform, oder aber gibt ein Feld mit mindestens 255 ganzzahligen Einträgen an. Diese Einträge dürfen Werte zwischen —128 und 127 annehmen und stellen die Amplitude der Welle dar. Der SOUND-Befehl legt die an ihn übergebenen Töne erst in einer Warteschlange ab und beginnt dann zu spielen. Dies kann man sich zunutze machen, um alle vier Kanäle zu synchronisieren. Dazu gibt man als ersten SOUND-Befehl SOUND WAIT an, danach die SOUND-Befehle für die gewünschten Kanäle. Nach der Angabe von SOUND RESUME werden die Kanäle zusammen wiedergegeben. Es soll nicht verschwiegen werden, daß die SOUND-Befehle sehr speicherplatzintensiv sind, doch durch das Warteschlangenkonzept muß die Übergabe der Töne auch nicht auf einmal erfolgen.

Jetzt wird es langsam Zeit auf den »aber«-Einwand von vorhin zurückzukommen: Das in Listing 3 wiedergegebene Programm übersteht zwar das erstmalige Starten anstandslos, läßt man es aber erneut mit dem RUN-Befehl laufen, »hängt« es sich ohne Fehlermeldung auf (ein Abbruch mit STOP ist jederzeit möglich). Die Ursache dieses Phänomens konnte bisher nicht geklärt werden.

Grafik und Animation

In den Grafikbefehlen liegen die eindeutigen Stärken des Amiga-Basic. Die vorhandenen Instruktionen sind komfortabel und leicht zu handhaben, so daß auch das Erstellen komplexerer Grafiken kein Problem sein sollte. In Listing 3 wird neben den erzeugten Tönen auch noch die Wellenform grafisch ausgegeben.

Der LINE-Befehl ist leistungsfähiger als auf den ersten Blick ersichtlich, denn mit ihm ist es nicht nur möglich, Linien zu zeichnen, sondern es können auch ausgemalte Rechtecke erzeugt werden. Daneben gibt es noch Befehle zum Setzen einzelner Punkte (PSET und PRESET), zum Zeichnen von Ellipsen (CIRCLE) und zum Zeichnen (AREA) und Ausmalen (AREAFILL) von geschlossenen Polygonzügen. PAINT füllt eine beliebige geschlossene Fläche mit einer wählbaren Farbe, während PATTERN bestimmt, mit welchem Muster Linien und Flächen gemalt werden.

Die voreingestellte Farbenzahl im Amiga-Basic beträgt 4. Will man mehrere Farben benutzen, so muß mit der SCREEN-Anweisung ein neuer Bildschirm geöffnet werden, wobei als Parameter die Auflösung und die Anzahl der Bitplanes (1 bis 5) übergeben werden. Die Anzahl der Farben errechnet sich aus 2^Anzahl der Bitplanes; also können auf einem Bildschirm maximal 32 Farben (inklusive Hintergrund) dargestellt werden. Zur Auswahl der Farben dient die PALETTE-Anweisung. Nach der Farbennummer (0 bis 31) werden jeweils durch eine Fließkommazahl zwischen 0 und 1 die Rot-, Grün- und Blauanteile angegeben. Eine wertvolle Hilfe bilden die Befehle GET und PUT. Sie lesen grafische Darstellungen vom Bildschirm als ganze Zahlen in ein Feld ein und umgekehrt.

Bilder nicht im IFF-Standard

Leider wurde das Amiga-Basic vor der Entwicklung des IFF-Standards geschrieben; daher sind die von den Befehlen erzeugten Datenformate zu den gängigen Malprogrammen nicht kompatibel. Listing 4 zeigt, wie einfach es ist, mit Amiga-Basic ansprechende Bilder zu erzeugen. Soviel zur stehenden Grafik. Wen ein ruhiges Bild langweilen sollte:

Die OBJECT-Befehle bringen Bewegung auf den Bildschirm. Auf dem Amiga gibt es zwei Arten von bewegungsfähigen Objekten: »Bobs« und die vom C 64 her bekannten Sprites. Bobs und Sprites unterscheiden sich in der Anzahl der möglichen Farben, der maximalen Größe und der Geschwindigkeit, mit der sie über den Bildschirm bewegt werden können. Da dem Amiga-Basic jedoch das Programm »Objedit« beiliegt, welches ein einfaches Editieren von Bobs und Sprites ermöglicht, ist es unnötig, sich hier über die Unterschiede Sorgen zu machen. Nach dem Erstellen der Objekte, ermöglichen es die OBJECT-Befehle, diese mit gleichförmiger Geschwindigkeit oder beschleunigt über den Bildschirm zu bewegen. Ebenso können Kollisionen registriert und verarbeitet, Objekte angehalten und gestartet sowie die Priorität gegenüber anderen Objekten verändert werden.

Unterprogramme und Libraries

Zählt man die definierbaren Funktionen mit, so kennt das Amiga-Basic drei Arten von Unterprogrammen. Zum einen die eben erwähnten Funktionen. Sie werden dem Basic durch DEF FN bekanntgemacht und mit FN aufgerufen. Eine Funktion darf nur eine Rechenvorschrift enthalten und ist somit auf eine Zeile beschränkt.

Eine wesentlich leistungsfähigere Art von Unterprogramm ist die Subroutine. Sie wird mit einem Label (zum Beispiel »Subroutine:«) begonnen und endet mit RETURN. Alle in einer Subroutine verwendeten Variablen sind global, das heißt sie sind nicht nur dem Unterprogramm bekannt, sondern auch dem Hauptprogramm. Die Subroutine wird mit dem GOSUB-Befehl aufgerufen.

Einer der wichtigsten Vorteile von Amiga-Basic sind jedoch die »echten« Unterprogramme. Das sind solche, bei denen die Variablen nur lokal, das heißt nur in diesem Unterprogramm gültig sind. Hierzu am besten ein Beispiel:

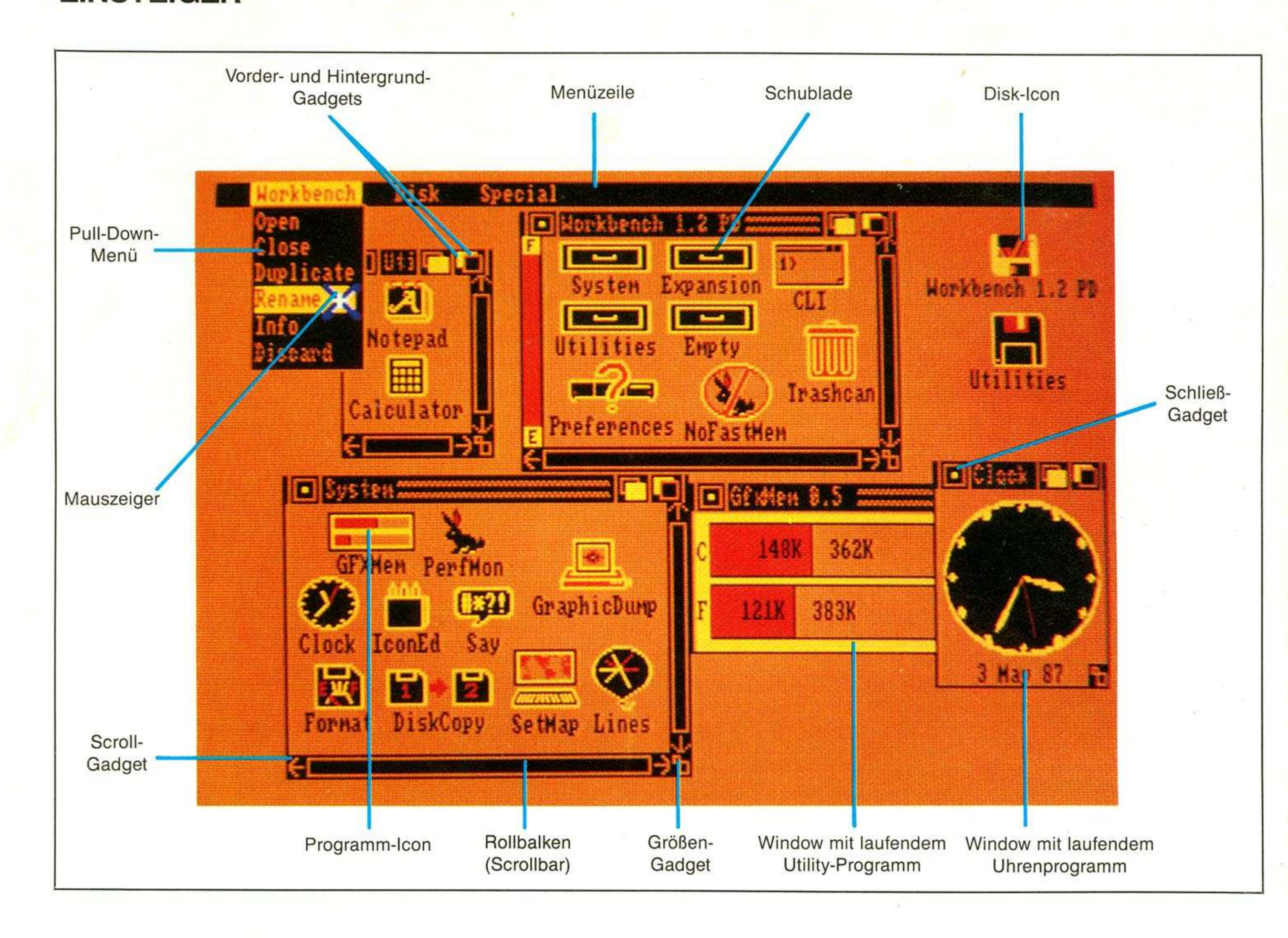
Angenommen im Hauptprogramm wird der Variable X irgendwann der Wert 1 zugewiesen; zugleich haben wir auch noch ein »echtes« Unterprogramm, und in diesem wird X = 2 gesetzt. Führt man jetzt im Hauptprogramm den Befehl PRINT X aus, so erhält man 1 — im Unterprogramm dagegen führt der Befehl PRINT X zum Ergebnis 2. Die beiden Variablen »wissen« also trotz gleichen Namens nichts voneinander, und dies ist ein entscheidender Vorteil: Sie können jetzt Unterprogramme sozusagen auf Vorrat schreiben, und sie bei Bedarf ohne Probleme aneinanderketten. Damit sind wir auch schon bei den Libraries (das sind im übertragenen Sinn Sammlungen von Unterprogrammen).

Unterprogramme müssen nicht unbedingt in Basic, sondern können auch in Maschinensprache geschrieben sein. Sie haben sogar die Möglichkeit, die Betriebssystemroutinen aufzurufen (vorausgesetzt, Sie wissen, welche Parameter Sie übergeben müssen). Allerdings gibt es da noch ein kleines Hindernis: Das Betriebssystem ist in der Sprache C geschrieben, und C verwendet eine andere Art der Parameterübergabe als das Basic. Aber wenn Sie sich schon gefragt haben, was das Programm »Convertfd« auf der Extra-Diskette soll: Es wandelt das Übergabeformat von C in das von Basic um und erzeugt die ».bmap«-Dateien, die Sie sicherlich auch schon in der »Basic-Demos«-Schublade gesehen haben.

Fazit

Von einer Ausnahme (Listing 3) abgesehen, konnte das Amiga-Basic positiv überraschen. Es liefert einen mächtigen Befehlssatz und ist durch die Unterstützung der Maus komfortabel zu bedienen. Profis werden es allerdings wegen seiner geringen Geschwindigkeit (Interpreter) und wegen der fehlenden Datentypen kaum für größere und zeitintensive Problemlösungen verwenden wollen. Hier ist, wie auch bei anderen Computern, die Maschinensprache oder ein Compiler vorzuziehen. Gewöhnungsbedürftig ist bei Amiga-Basic die Tatsache, daß Programme ohne Zeilennummern geschrieben werden können. Diese Eigenschaft erinnert stark an Pascal und fördert die strukturierte Programmierung. Durch die hohe Kompatibilität des Amiga-Basic zu GW-Basic stellt das Umschreiben von Programmen für den Anwender kein großes Problem dar.

(J.K. Singer/ks)



Amiga-Computerlexikon für Einsteiger

Als Computeranfänger oder Umsteiger von einem anderen Computer stößt der Amiga-Anwender ständig auf neue Begriffe, die hier einmal verständlich und ausführlich erklärt werden sollen.

Die grafische Benutzeroberfläche des Amiga ist zweifellos einfach zu bedienen, doch auch der Umgang damit will gelernt sein. Immer wieder tauchen Begriffe auf, die einen Einsteiger vor Probleme stellen. Wir sehen es als Aufgabe und Verpflichtung gegenüber unseren Lesern, hier konkrete Hilfestellung zu leisten.

Die erste Frage, die sich ein »Computerneuling« wohl stellen wird, lautet natürlich: Was ist überhaupt eine »Benutzeroberfläche«? Prinzipiell handelt es sich hierbei um eine Schnittstelle, also ein Bindeglied zwischen Mensch und Computer. Bei Personal Computern kennt man hauptsächlich textorientierte Benutzeroberflächen, das heißt der Computer reagiert auf die Eingabe von Befehlen in alphanumerischer Form. Eine grafische Benutzeroberfläche stellt statt alphanumerischen Befehlen grafische Elemente zur Verfügung; diese sind für den der Fachsprache Unkundigen wesentlich verständlicher, genauso wie in den Sportstadien der Welt die international gültigen Pictogramme unlesbare, fremdsprachige Schriften abgelöst haben.

Beim Amiga heißt die grafische Benutzeroberfläche »Intuition«, was sich in etwa mit »gefühlsmäßigem Handeln« oder auch »Anschauliches« übersetzen läßt. Die Philosophie, die dahinter steckt, ist denkbar einfach: Der Anwender soll nicht mit überflüssigen Details belastet werden.

Nun, ganz so einfach geht es leider auch wieder nicht, denn mit Fachbegriffen wird man auch hier regelrecht überflutet. Dazu kommt noch, daß sich die Software-Entwickler noch nicht auf Standards in der Übersetzung der Fachbegriffe geeinigt haben.

Es ist teilweise auch sehr schwierig, eine sinngemäße Übersetzung zu finden. Ein Beispiel: Die wörtliche Übersetzung von »Gadget« ist »Dingsbums«, was wirklich nicht besonders viel aussagt. Da ein Gadget aber sowohl ein Symbol als auch ein Symbolfeld oder ein Symbolfeld mit Schalterfunktion sein kann, ist es nicht einfach, eine wirklich aussagekräftige Übersetzung zu finden. Wir machen uns diesbezüglich aber auf jeden Fall noch Gedanken, um sowohl für die dem »Computerchinesisch« skeptisch gegenüberstehenden Anwender als auch für die Freaks eine gleichermaßen akzeptable Lösung zu finden. Es ist aber zumindest sinnvoll, immer den englischen Originalausdruck im Hinterkopf zu behalten.

Der Schreibtisch des Amiga

Intuition stellt dem Benutzer einen Schreibtisch (Workbench) zur Verfügung, auf dem er mit vielen symbolhaften Zeichen (Icons, Gadgets) und frei

GO AMIGA!

	sFr.	DM
ABSOFT		
AC Fortran	590	649
AC Basic	390	429
ACCESS SOFTWARE		
*Tenth Frame	79	89
ACCESS ASSOCIATES		
Alegra 512K RAM	759	879
Alegra 2M RAM	1499	1649
Autopal Expander	68	70
ACCOLADE		
Mean 18 Golf	59	69
ACTIVISION		
Hacker I	49	59
Borrowed Time	49	59
Mindshadow	49	59
Shanghai	69	79
Portal	79	89
Ch.ship Golf	69	79
Ch.ship Basketball	69	79
Ch.ship Baseball	69	79
Ch.ship Football	79	89
*Write Choice Elite	159	179
AEGIS DEVELOPMENT		
Aegis Draw	249	249
DEMO Draw	15	15
Aegis Draw plus	499	499
Aegis Animator	279	279
DEMO Animator	15	15
Aegis Images	79	89
Aegis Art Pak # 1	59	69
Aegis Impact	179	199
Aegis Sonix	159	189
*Aegis Diga!	159	189
*Arozok's Tomb	99	119
AMISOFT		
Diskzine Nr. AA–AG	19	29
ANAKIN RESEARCH		
The Easel	899	999
DEMO Easel	15	15
APPLIED VISIONS		
Future Sound	350	399
ARRAY's INC.		
Master AM DOS Buch	39	49
ARTWORX		
Strippoker	69	79
ASSOCIATED COMPUTER		
Brush Works	69	79
BANTAM		
AM Manual Buch	59	69
Express DOS Refdisk	59	69
AM User Guide Buch	49	59
BAUDVILLE		
Video Vegas	59	69
BETHESDA SOFTWORKS		
Gridirion	119	139
DEMO Gridirion	15	15
BROWN WAGH		
Zuma Fonts	59	69
DEMO Zuma Fonts	15	15
TV Text	199	249
DEMO TV Text	15	15
C LTD.		
aMEGA 1 M RAM	999	1100
HARDDISK 20 M	2000	2200
TIMESAVER	159	179
CENTRAL COAST SOFTWARE		
DOS 2 DOS	118	128
*Disk 2 Disk	99	119
CLASSIC IMAGE		
Diablo	59	69
COMMODORE		
AmigaTerm	99	119
Enhancer		
1,2 KS/WB	30	40
AM Jahrbuch 1987	10	10
COMPUMED		
Mirror Koppierer	99	119
Mirror Hacker Pak	99	119
COSMI		
Super Huey	59	69
CREATIVE MICRO		
KICK ELIMINATOR	259	289
DIAMOND SOFTWARE		
*Investor	1499	1699
DEMO Investor	15	15
C monitor V 200	179	199
DIGITAL CREATIONS		
Gizmos 2.0	139	159
*D'Buddy	159	179
Digital Link	139	159
DIGITAL SOLUTIONS		
*LPD Writer	259	289
*LPD Filer	259	289
*LPD Planner	259	289
DISCOVERY		
Marauder II	79	89
Grabbit	59	69
ECE RESEARCH		
ECE Midi Interface	119	139
Amiga DOS Ref Card	6	7
*Speedy Amigos	129	139
ELECTRONIC ARTS		
Deluxe Paint 2	199	249
Paint Data 1	59	69
Paint Data 2	59	69
Paint Data 3	59	69
Deluxe Print	119	139
*Deluxe Video 2	199	249
Deluxe Music	199	249
Music Data 1	59	69
Articfox	59	69
Adventure Construct	66	79
*Demo Adventure	15	15
Marble Madness	59	69
Chessmaster	79	89
Bards Tale	89	99
*Demo Bards Tale	15	15
EPYX		
Wintergames	59	69
FAUG		
RoboCity Zeitschrift	4	5
Jahresabonnement	42	54
GOLD DISC INC.		
Pagesetter	299	349
Gold Spell	89	99
*Lazerscript	89	99
GOLDEN HAWK		
GoldMidi Interface	159	179
GRAFOX OF ENGLAND		
Logistic englisch	299	329
IDEAS CREATED		
Filter für HI-Res	29	39
IMPULSE		
*Prism	139	159
*Proclip	59	69
INFINITY SOFTWARE		
Grand Slam Tennis	69	79
*Galileo	199	219
INNOVATIVE TECHNOLOGY		
Diskmappe 3 1/2	19	24
Diskmappe 5 1/4	19	24
INOVATRONICS		
Power Windows	179	199
INTERACTIVE SOFTWORKS		
*The Calligrapher	199	219
ISM INC.		
The Surgeon	119	129
JAGHWARE INC.		
*Alien Fires	79	89
JUMPDISK		
Diskmagazin	19	29
LATTICE		
S Compiler V 3.1	450	450
Unicalc	179	179
Mac Library	199	199
dBC III	299	299
MANX FOFTWARE		
C Commercial	999	1149
C Developers	599	699
*C Personal	399	449
MARKT + TECHNIK		
Superbase	199	249
MERIDIAM SOFTWARE		
Zing CLI Deluxe	159	179
DEMO Zing	15	15
*Games Gallery	59	69
*Zing Keys	99	119
METACOMCO		
Shell CLI Deluxe	99	119
Assembler	159	179
Toolkit	89	99
MICHIGAN SOFTWARE		
Insider – 1 M RAM		
interne Expansion	699	769
MICRO ILLUSIONS		
Dynamic CAD	999	1149
DEMO Dynamic CAD	15	15
*Fairy Tale Adv.	99	119
MICROPOSE		
*Gunship	79	89
MICROSMITHS		
TxEd	79	89
Fast Fonts	79	89
MICROSOFT PRESS		
The Amiga Buch	59	69
MICRO SYSTEMS SOFTWARE		
Analyze V 2.0	299	349
BBS Bulletin Board	199	229
Online	139	159
Scribble plus	199	229
Organize	199	229
MILES COMPUTING		
Quintette	89	99
MIMETICS (SOUNDSCAPE)		
Pro Studio	299	349
Midi Interface	99	119
Sound Sampler	199	229
MINDSCAPE		
Deja Vue	79	89
Univited	79	89
Halley Project	59	69
Balance of Power	79	89
Defenders of Crown	79	89
Sindbad	79	89
SDI	79	89
*King of Chikago	89	99
Perfect Score	159	179
Mastertype	79	89
NEW HORIZONTS		
Flow	199	229
DEMO Flow	15	15
Pro Write	259	289
NEW-TEK		
Digi View	399	459
DEMO Digi View HAM	15	15
*Digi Paint	119	129
NORTHEAST SOFTWARE		
*Publisher	399	459
*Organizer	99	119
ORIGIN SYSTEMS		
Ogre	69	79
Ultima 3	120	135
*Ultima 4	128	135
*Autoduel	99	119
OTHER GUYS		
Omega File	159	179
Great States	79	89
Promise	99	119
PACIFIC CYPRESS		
Expander 2M RAM	1399	1599
PARCTECH		
Kabel Parallel	39	49
Kabel RS 232	29	39
PIM PUBLICATIONS		
Amazing Zeitschrift	7	8
Jahresabonnement	72	84
PRECISION SOFTWARE		
Amiga System Buch	39	49
PROGRESSIVE PERIPHERALS		
CLI Mate	69	79
PSYGNOSIS		
Brataccas + Arena	79	89
*Barbarian	69	79
*Terrorpods	69	79
RAINBIRD		
Jewels of Darkness	39	49
SIERRA ONLINE		
Kings Quest 1	89	99
Kings Quest 2	89	99
Kings Quest 3	89	99
SLIPPED DISK		
BASIC		
Tastaturschablone	19	25
DOS Tastaturschablone	19	25
SOFTWARELAND		
jede Disk Public	15	15
10er Pack Public	135	135
GO AMIGA! Datei	179	199
DEMO Datei	15	15
*GO AMIGA! Text	179	199
SSI		
Computer Baseball	59	69
SUBLOGIC		
Flightsimulator 2	99	119
*Jet	99	119
*Scenery Disk	49	59
SUNRIZE INC.		
Perfect Sound	159	179
DEMO Perfect Sound	15	15
SWISS COMPUTER ARTS		
*Deluxe		
Titelkonstruktionsset	79	89
DEMO Titelset	15	15
TDI SOFTWARE		
Modula-2 Standard	199	219
Modula-2 Developer	299	349
Modula-2 Commercial	599	699
DEMO Modula-2	15	15
Examples Disk	49	59
Amiga Editor	99	119
The Grid	99	119
Kermit Source	49	59
TAURUS IMPEX		
Acquisition	599	649
DEMO Acquisition	30	30
TRUE BASIC		
Interpreter	299	349
Compiler Runtime	299	349
Developers Toolkit	99	119
Libraries:		
Sorting + Searching	99	119
Advanced Strings	99	119
3 D Graphics		119
Programme:		
Algebra 1	99	119
Algebra 2	99	119
Calculus	99	119
Discrete Maths	99	119
Probability Theorie	99	119
Trigonometry	99	119
Pre Calculus	99	119
TrueStat	159	179
UNISON WORLD		
Printmaster plus	99	119
Art Gallery 1	59	69
Art Gallery 2	59	69
VERSASOFT		
dBMan	299	349
DEMO dBMan	15	15
VIZA SOFTWARE		
*Vizawrite englisch	299	359
DEMO englisch	15	15
*Vizawrite deutsch	398	498
VIP TECHNOLOGIES		
VIP Prof englisch	299	329
WAVE PAD		
Mouse Pad	12	15
WESTCOM INDUSTRIES		
ADFO		
Disk File Organizer	99	119
HardHat		
Harddisk Backup	139	159

* = Lieferbar nach Verfügbarkeit.

Preisänderungen vorbehalten.

Bestellservice:

BRD: 0041-1-3115959
CH: 01-3115959

Geschäftszeiten:
10.00–12.30, 13.30–18.30 Uhr, außer montags,
Sa.: 10.00–16.00 Uhr.

Versand ins Ausland nur Vorkasse (Scheck, bar, Visa Card, Master Card) zzgl. DM 7,– Porto.

softwareland
Franklinstraße 27
CH-8050 Zürich (Schweiz)

verschiebbaren Fenstern (Windows) durch leicht einprägbare Aktionen seinen Amiga bedienen kann. Eben fast wie auf einem Schreibtisch, auf dem man seine Arbeitsblätter, Radiergummis und Stifte benutzt.

»Gadgets«, »Requesters«, »Mauszeiger«, »Windows«, »Screens« und »Icons« heißen die verschiedenen Grafikelemente von Intuition, die die Mensch-Computer-Kommunikation vereinfachen sollen.

Windows, also Fenster, öffnen sich beim »Anklicken« eines Symbols (Icon). Fenster stellen den Inhalt einer Diskette oder eines Teils daraus dar; die Diskette oder Teile davon werden durch ein Symbol (Icon) abgebildet.

Es werden »Disk-Symbole«, die die Diskette repräsentieren, und »Schubladen« (englisch »Drawers«) für Unter-Inhaltsverzeichnisse (Subdirectories) sowie Programm-Symbole für einzelne Programme unterschieden. Das »Anklicken« (also zweimaliges Drücken auf den linken Mausknopf) mit dem Mauszeiger bewirkt je nach Icon-Typ das Starten eines Programms oder das Öffnen eines Windows — manchmal auch gar nichts, wenn es sich zum Beispiel um ein Datenfile ohne Hauptprogramm handelt. Bei manchen Icons wird durch das Anklicken nämlich nicht das Datenfile geladen, sondern ein dazugehöriges Hauptprogramm. Wenn Sie sich mit »Info« im Workbench-Menü ein Deluxe-Paint-Bild ansehen (selektieren Sie das Bild, indem Sie nur einmal auf das Icon klicken), steht dort unter »Default-Tool« das Programm, das beim Anklicken des Icons gestartet wird. Erst dieses Programm lädt endgültig das angeklickte Datenfile.

Symbole der Workbench

Ein Symbol (Icon) repräsentiert also etwas. Intuition kennt allerdings auch noch andere Elemente, die man anklicken kann: die sogenannten »Gadgets«. Im Gegensatz zu den Icons repräsentieren sie aber keine Disketten oder Files, sondern üben Kontrollfunktionen aus. Das heißt beim Anklicken eines Gadgets passiert etwas mit den Elementen der Benutzeroberfläche.

Intuition kennt mehrere Standard-Gadgets; für Programmierer steht zudem die Möglichkeit offen, eigene Gadgets zu entwickeln.

Das meistgenutzte Gadget ist in der Regel das »Close-Gadget« oder auf deutsch »Schließschalter«. Er befindet sich im Normalfall in der linken oberen Ecke eines Windows. Ein Mausklick auf dieses Gadget genügt, um ein Window zu schließen oder gleichzeitig aus einem Programm auszusteigen (wenn das Programm es vorsieht). Das »Size-Gadget« oder »Größensymbol« (frei übersetzt) befindet sich rechts unten und ist für eine Vergrößerung oder Verkleinerung eines Fensters zuständig (Mauszeiger auf das Größensymbol positionieren und bei gedrücktem linken Mausknopf die Größe verändern, dann loslassen). Ist ein Fenster so klein, daß es nicht alle dazugehörigen Symbole (Icons) auf diesem kleinen Raum anzeigen kann, läßt sich der Ausschnitt des Arbeitsblatts, der durch das Fenster dargestellt wird, verschieben. Dazu dienen die sogenannten »Scroll-Gadgets«; ein Klick auf solch ein Gadget verschiebt den Fenster-Inhalt um die Anzahl der Bildpunkte, die ein Icon hat. Statt dieser ruckweisen Verschiebung geht es auch stufenlos. Dazu stellt uns Intuition sogenannte »Scroll-Bars« zur Verfügung. »Bar« bedeutet Balken, die Standard-Übersetzung heißt »Rolleiste« oder »Rollbalken«. Dieser zeigt an, wieviel des tatsächlichen Window-Inhalts im Ausschnitt zu sehen ist. Der gesamte Balken gibt die Gesamtgröße an und das Rechteck den sichtbaren Teil. Um sich den weiteren Inhalt anzusehen, brauchen Sie nur das Rechteck zu nehmen und im Kasten zu verschieben. Klicken Sie statt dessen auf den freien Bereich neben dem Rechteck, springt der Inhalt um eine Ausschnittgröße zur Seite. Wenn Ihnen das zuviel ist, können Sie auch die Pfeile nach links oder rechts beziehungsweise oben oder unten anwählen. Sie bewirken eine Verschiebung um einen halben Ausschnitt. So verfügen Sie über insgesamt drei Wege, um in einem verkleinerten Fenster sich den gesamten Inhalt anzusehen, ohne das Fenster selbst mit dem »Size-Gadget« vergrößern zu müssen.

Einen weiteren Balken sehen wir in der obersten Zeile des Fensters. Dort steht der Fenster- oder Programmname, der Rest der »Zeile« wird mit Strichen ausgefüllt. Diesen Balken nennt man auch »Drag-Bar« oder »Verschiebebalken«. Die Intuition-Balken fallen aus rein programmiertechnischen Gründen übrigens auch unter die Gadgets. Derer gibt es natürlich noch mehr, genau gesagt zwei: das »Front-Gadget« (sinngemäß: »Vordergrundschalter«) und das »Back-Gadget« (Hintergrundschalter).

Gadgets haben viele Gesichter

Beide Gadgets sind zur Kontrolle der Vorder- oder Hintergrundpriorität der Fenster untereinander vorgesehen. Andere Computer mit grafischer Benutzeroberfläche kennen übrigens diese Art der Gadgets in der Regel nicht; hier wird einfach der sichtbare Teil eines Windows angeklickt, und schon ist es im Vordergrund. Natürlich ist es ungünstig, wenn ein großes Fenster ein kleines Fenster vollkommen überdeckt, da nun nach den verdeckten Gadgets gesucht werden muß.

Intuition-Elemente wie die Fenster sind nichts anderes als grafische Repräsentationen von Arbeitsblättern oder Schubladen. Ein Window-Typ erscheint nur, wenn ein Disk-Icon angeklickt wird. Dieser Window-Typ enthält an der linken Seite einen Balken, der anzeigt, wie voll die Diskette ist. (Kein Gadget, weil der Balken keine Kontrollfunktion ausübt!)

Fenster (Windows) sind wiederum nur Teile von »Screens«. Die »Screens« sind etwas ganz Amiga-Spezifisches und werden durch den Coprozessor »Copper« ermöglicht. Screens sind eine Art »virtuelle Bildschirme«; durch die speziellen Fähigkeiten des Amiga (Multitasking) können Programme so abgearbeitet werden, als hätte jedes einen kompletten Bildschirm zur Verfügung. Der Benutzer kann nun diese Screens nach Belieben auf seinem »echten« Bildschirm verteilen und so die Teile eines jeden Screens sehen, die er sehen will. Das Verschieben funktioniert mit Hilfe des oberen Balkens, auch dieser Balken wird »Drag-Bar« genannt. Der Verschiebebalken enthält, wie auch die Windows, ganz rechts die Back- und Front-Gadgets (Vorder- und Hintergrundschalter), die die Priorität (was liegt vorne, was liegt hinten) der virtuellen Bildschirme regeln. Auch die Workbench ist ein solcher Screen und wird deshalb oft »Workbench-Screen« genannt.

Der obere Balken des Screens hat zusammen mit dem linken Mausknopf also die Funktion eines Verschiebebalkens; mit dem rechten Mausknopf wird er zur Menüzeile. Dort stehen jetzt verschiedene Optionen zur Verfügung, in der Workbench sind das »Workbench«, »Disk« und »Special«. Bewegen Sie bei gedrücktem rechten Mausknopf den Mauszeiger auf eine dieser Optionen, erscheint ein »Pull-Down-Menü«. »Pull Down« heißt »herunterziehen« oder »abrollen«. Führen Sie Ihren Mauszeiger entlang der angebotenen Menüpunkte, so erscheint ein Balken auf dem Bildschirm. Der farblich unterlegte Menüpunkt zeigt, welche Funktion ausgeführt wird, wenn Sie den Mausknopf loslassen.

Die meisten dieser Menüfunktionen setzen voraus, daß vorher ein Icon selektiert wurde, an dem die Operation durchgeführt werden soll. Dies geschieht durch einmaliges Anklicken des Symbols mit dem Mauszeiger.

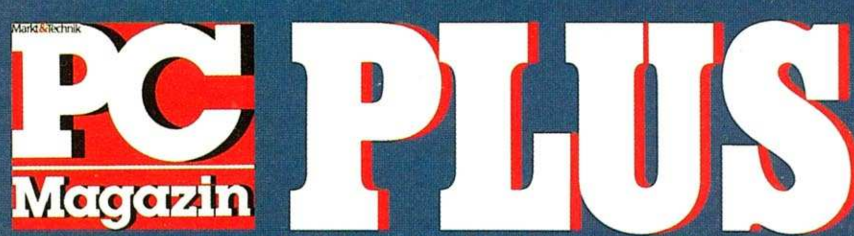
Intuition kennt außerdem sogenannte »Requester«. Englisch »Request« steht für »Frage«. Ein Requester tritt immer dann in Aktion, wenn das Programm etwas Wichtiges erfahren will. In unserem Fall ist das Programm die Workbench. Ziehen Sie einmal die Workbench-Diskette aus dem Laufwerk und klicken dann ein Icon an. Ganz klar, der Computer kann das zugehörige Programm nicht laden; links oben im Screen erscheint die sogenannte »Requester-Box«, in der die Fragerequester »Retry« (Versuch wiederholen) und »Cancel« (Abbrechen) erscheinen. Durch Anklicken der Felder entscheidet man, ob der Ladevorgang wiederholt oder abgebrochen wird.

Das Programm fragt nach

Die berüchtigte »Guru Meditation« als Zeichen eines Systemabsturzes ist ebenfalls ein Teil der Benutzeroberfläche »Intuition«. Diese Art der Meldung nennt sich »Alert« (englisch für Alarm) und kann auch von anderen Programmen aus — zum Beispiel als Warnung für zu wenig Speicher — verwendet werden. Beim »Guru Meditation Alert« ist der Amiga ausgestiegen. Wir tun es auch, in dem Wissen, nötige Grundlagen für Ihre weitere Arbeit vermittelt zu haben.

(Manfred Kohlen/pd)





7/8-'87 DAS GROSSE PERSONALCOMPUTER-MAGAZIN

Unermeßliche Bereiche einer faszinierenden Computerwelt entdecken Sie durch das PC-Magazin-Plus. Sie lernen eine ganz neue Welt der IBM-PCs und kompatibler Systeme mit überraschenden Perspektiven und Möglichkeiten kennen — beim Programmieren, bei Text- und Datenverarbeitung, Grafik und Homeentertainment.

Mit einer Vielzahl anspruchsvoller Listings und Anwendungen, Programmierkursen, Tests, Ratschlägen und Bauanleitungen zur Hardwareerweiterung beherrschen Sie schnell Ihren PC und nutzen seine Leistung optimal aus.

PC-Magazin-Plus, die ganz neue PC-Zeitschrift für alle, die IBM-PC und Kompatible mit Engagement benutzen oder einsetzen wollen — ob Einsteiger oder Profi. Sie erhalten sie ab 16. Juni bei Ihrem Zeitschriftenhändler.

Sichern Sie sich jetzt Ihre persönliche PC-Zukunft.

Kennenlern-Angebot

mit einem kostenlosen Probeexemplar PC-Magazin-Plus

Ja, ich interessiere mich für PC-Magazin-Plus und möchte ein kostenloses Probeexemplar dieser Zeitschrift. Wenn ich PC-Magazin-Plus weiterlesen will, brauche ich nichts zu tun, ich bekomme dann PC-Magazin-Plus regelmäßig per Post zum günstigen Jahrespreis von 84,— DM (für 12 Ausgaben, Auslandspreise und Studentenabo siehe Impressum).

Geld-zurück-Garantie:

Ich kann das Abonnement jederzeit kündigen, es gibt keine Kündigungsfrist. Zuviel bezahlte Beträge erhalte ich zurück.

Name ____________ Vorname ____________

Straße ____________ PLZ/Wohnort ____________

Telefon ____________ Datum, 1. Unterschrift ____________

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Besteladresse Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Str. 2, 8013 Haar bei München widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs an obige Adresse. Ich bestätige dies durch meine 2. Unterschrift.

Datum, 2. Unterschrift ____________

AMIGA TEST

Professionelle Dateiver

**Datenbanken sind für den Amiga momen-
fessionellen Programmen zählen Super-
testet, ob sie halten können, was sie**

Der Amiga wird gelegentlich als Spiel-Computer bezeichnet. Daß man ihn aber auch ohne teure Erweiterungen sinnvoll nutzen kann, zeigen die Programme »Superbase« und »Softwood File II«. Beide laufen auf der 512-KByte-Version des Amiga und sind zur Not auch mit einem einzelnen Diskettenlaufwerk zu verwenden. Bevor wir uns mit Superbase Amiga beschäftigen, das Besitzern eines Commodore-Computers sicher zumindest namentlich ein Begriff ist, wenden wir uns zunächst Softwood File II zu.

Dieses Datenbankprogramm wird auf Diskette mit einem englischen Handbuch ausgeliefert. Die Autostart-Diskette enthält neben dem Programm und den üblichen Workbench-Tools auch drei Demodateien. Diese sind jedoch noch völlig auf Amiga Kickstart 1.1 bezogen. Wer die Version 1.2 des Betriebssystems verwendet, sollte Softwood File II nicht booten, sondern von der Workbench aus starten. Beim Starten von Softwood File II fällt sofort ein lobenswerter Punkt ins Auge. Das Programm fragt, wieviel Prozent vom verfügbaren Hauptspeicher es nutzen soll. Auch extern angeschlossenes Fast-RAM wird berücksichtigt (Bild 1). So wird der Amiga bei größerem Speicherausbau auch wirklich für Multitasking freigehalten. In der 512-KByte-Version belegt das Programm aber selbst schon soviel Speicher, daß nur noch etwa 70 KByte für Daten zur Verfügung stehen.

Der Aufbau einer Datenbankstruktur ist bei Softwood File II erfreulich einfach. Man kann bei Eingabefeldern zwischen Textdaten, numerischen Werten, Datumeingabe und auch »Ja/Nein«-Abfragen unterscheiden. Sogar Telefonnummern und Zeitangaben lassen sich als Eingabewerte anwählen. Wenn Sie zum Beispiel Zinssätze errechnen wollen, definieren Sie ein Feld, in dem nach einer beliebigen mathematischen Formel Werte aus anderen Feldern mit den Grundrechenarten verknüpft und dann dargestellt werden. Softwood File II erledigt nach Bedarf auch kleinere Rechnungen automatisch. Sollten Sie hinterher feststellen, daß ein bestimmtes Feld in Ihrer Datei nicht mehr benötigt wird, löschen Sie es mit dem Menüpunkt »DELETE«.

Die Bildschirmmaske wird geändert, indem man mit der Maus ein bestimmtes Eingabefeld anklickt und wie ein Workbench-Icon verschiebt. Auch die Längen der verschiedenen Eingabezeilen werden so mit der Maus ganz einfach geändert. Die Datensätze mit Ihren Feldern sind entweder einzeln in der vorgewählten Bildschirmmaske oder spaltenweise wie bei einer Finanzbuchhaltung darstellbar (Bild 2). Hauptsächlich verwendet man in der Praxis wohl die Spaltendarstellung, da viele Datensätze gleichzeitig angezeigt werden und man sie mit der Maus einfach anklickt, wenn man sie braucht. Sollten Sie in der Spaltendarstellung einer bestimmten Spalte eine andere Position oder Breite zuweisen, benutzen Sie dazu ebenfalls die Maus. Für fast jede Anwahl und Steuerung wird die Maus verwendet. Wer das Arbeiten mit der Maus ablehnt oder als ungewohnt und umständlich empfindet, kann für die wichtigsten Programmpunkte oft auch die Tastatur verwenden. Die Tasten <RETURN> oder <ENTER> wirken zum Beispiel meist wie ein Klick mit der Maus auf ein »OK«-Gadget. Für eine flüssige Eingabe von Daten ist das sehr nützlich. Ein Schwachpunkt dabei ist jedoch, daß man für jeden Datensatz und auch für jede andere Eingabe den Eingabebefehl neu geben muß. Besser wäre eine fortlaufende Eingabe, die nur bei Bedarf abgebrochen wird.

Datenausgabe mit Format

Neben der Definition der Bildschirmmaske mit der Maus, ist bei den Daten rechtsbündige, zentrierte und auch linksbündige Datenausgabe einstellbar. Sie können auch ganze Spalten verdecken, damit vielleicht Preise im Ausdruck nicht sichtbar sind. Bei Spalten, in denen Zahlen oder ein Datum steht, kann das Ausgabeformat auf unterschiedliche Weise geändert werden. Verschiedene Währungssymbole stehen zur Verfügung und auch die Schreibweise des Datums wird durch einmaliges Einstellen an europäische Verhältnisse angepaßt. Die aktuellen Definitionen für die Anzeige und den Druck lassen sich als »VIEW« getrennt von den Daten speichern. Das ist keine Spielerei, sondern hilfreich, wenn Sie dieselben Datenmengen in verschiedener Form ausdrucken oder anzeigen wollen.

Will man die Datensätze nach einem bestimmten Feld sortieren, wird in der Spaltendarstellung der jeweilige Spaltenkopf angeklickt und über einen der zwei Menüpunkte in auf- beziehungsweise absteigender Reihenfolge sortiert. Mit dem Menüpunkt »Progressive« kann nun zusätzlich nach anderen Spalten sortiert werden. Wenn Sie beispielsweise eine Adreßdatei nach dem Nachnamen sortiert haben, können Sie diese Sortierung beibehalten und innerhalb dieser Vorsortierung alle Vornamen alphabetisch ordnen lassen. Wollen Sie wieder nur nach einer Spalte sortieren, wählen Sie im »Sort«-Menü den Punkt »Single Column« an. Vorsortierungen werden damit wieder aufgehoben. Störend ist, daß nach jedem Sortiervorgang die jeweilige Spalte neu angeklickt werden muß. Schnelles Testsortieren einer Spalte ist somit kaum möglich.

Die Abfragemöglichkeiten, die eine gute Datenbank aus-

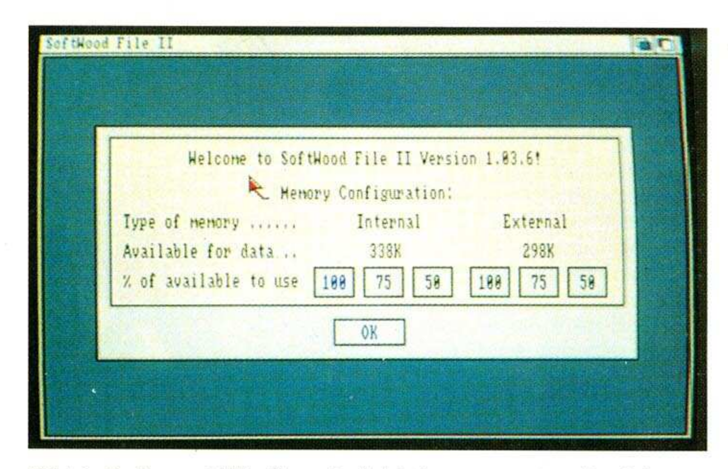


Bild 1. Softwood File II muß nicht den gesamten Speicher Ihres Amiga belegen

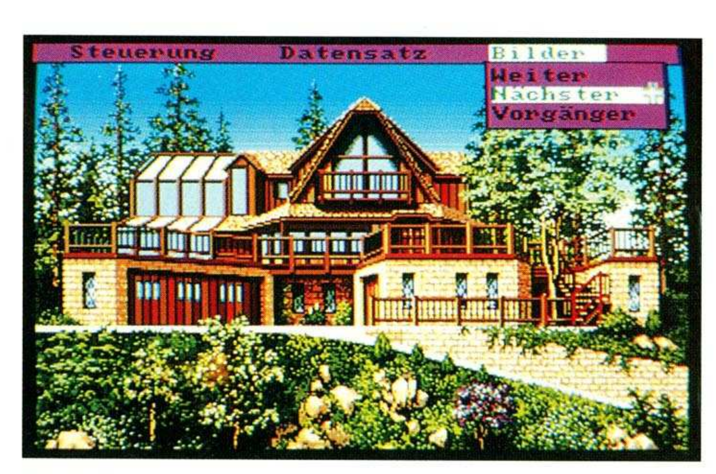


Bild 3. Superbase Amiga kann auf Wunsch auch Bilder als externe Dateien verwalten

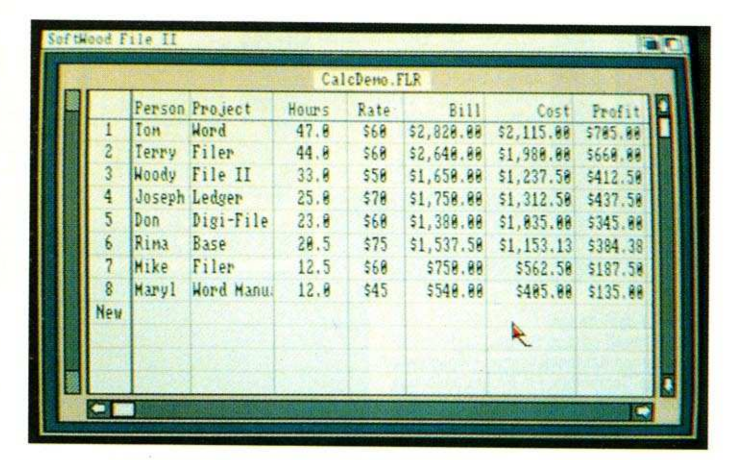
waltung auf dem Amiga

tan noch Mangelware. Zu den ersten probase und Softwood File II. Wir haben geihrem ersten Eindruck nach versprechen.

zeichnen, sind in Softwood File II leider etwas dürftig ausgefallen. Die üblichen Filtermöglichkeiten für Gleichheit, Ungleichheit, größer oder kleiner sind zwar vorhanden, aber man kann Sie nur jeweils auf die gerade angewählte Spalte anwenden. Dabei entscheidet man, ob sich die nächste Suche auf alle Datensätze oder nur auf die zuletzt ausgewählten bezieht. Um eine Auswahl zu treffen, muß man jede einzelne Selektion für sich durchführen. Das ist schon bei geringen Datenmengen eine mühsame Angelegenheit. Positiv ist aber, daß es die Funktion » * « gibt. Wenn Sie Textfelder auswählen, brauchen Sie den Suchbegriff nicht auszuschreiben. Es genügt, wenn Ihr Suchbegriff einen Teil eines Feldinhalts ausmacht, um einen Datensatz auszuwählen. Sie können so zum Beispiel alle Datensätze selektieren, die in einem bestimmten Feld ein »ller« enthalten, um alle »Müller« in den verschiedenen Schreibweisen zu erfassen. Auch Groß-/Kleinschreibung wird dabei ignoriert.

Softwood File II stellt Ihnen im »Option«-Menü einige Funktionen zur Verfügung, die bei der Datenbearbeitung hilfreich sein sollen. Es gibt die Funktion »Auto Capitalize«, durch die bei Eingaben der erste Buchstabe automatisch groß geschrieben wird. Die eigentlich gute Idee ist aber in vielen Fällen nutzlos, denn es wird nicht der erste Buchstabe des Feldinhalts groß geschrieben, sondern der erste Buchstabe, den man im Feld über die Tastatur eingibt. Das Programm kümmert sich nicht darum, ob es auch der erste Buchstabe des eingegebenen Wortes ist, oder ob man nur eine Eingabe korrigiert.

Eine sinnvolle Funktion ist dagegen »Grid On/Off«. Damit legen Sie ein Netz von Hilfslinien über die Spaltendarstellung, durch die alle Daten voneinander getrennt werden und viel leichter lesbar sind.



SoftWood File II — CalcDemo.FLR

	Person	Project	Hours	Rate	Bill	Cost	Profit
1	Tom	Word	47.0	$60	$2,820.00	$2,115.00	$705.00
2	Terry	Filer	44.0	$60	$2,640.00	$1,980.00	$660.00
3	Woody	File II	33.0	$50	$1,650.00	$1,237.50	$412.50
4	Joseph	Ledger	25.0	$70	$1,750.00	$1,312.50	$437.50
5	Don	Digi-File	23.0	$60	$1,380.00	$1,035.00	$345.00
6	Rima	Base	20.5	$75	$1,537.50	$1,153.13	$384.38
7	Mike	Filer	12.5	$60	$750.00	$562.50	$187.50
8	Maryl	Word Manu	12.0	$45	$540.00	$405.00	$135.00
New							

Bild 2. Die Daten werden bei Softwood File II sehr übersichtlich angezeigt



Bild 4. Der Filter zum Suchen von Datensätzen ist bei Superbase sehr vielseitig

Übersichtlicher Aufbau

Für die Dateneingabe nützlich ist auch die Funktion »Repeat Data Entry«. Ist diese Funktion aktiviert, können Sie bei der Eingabe eines Datensatzes ein leeres Feld mit dem Inhalt des zuvor eingegebenen Datensatzes füllen, indem Sie nur <RETURN> drücken. Beim Schreiben von Listen mit teilweise gleichem Inhalt ist dies eine sehr begrüßenswerte Erleichterung.

Der Datenaustausch mit anderen Programmen wird durch zwei Funktionen unterstützt, mit denen eine ASCII-Datei eingelesen — und auf diese Weise Datensätze integriert werden können — oder der aktuelle Dateninhalt von Softwood File II als ASCII-Datei auf Diskette geschrieben werden kann.

Bevor Daten ausgegeben werden, muß man in den Preferences der Workbench und in Softwood File II das Papierformat angeben. Da das nur einmal bei der Druckerinstallation nötig ist, fragt man sich, weshalb diese doppelte Einstellung vorgenommen werden muß. Wenn das Druckformat ausgewählt ist, wird der Menüpunkt »To Printer« angewählt und alle Daten lassen sich ausdrucken. Verdeckte Spalten und anderes, was im aktuellen »View« der Daten definiert ist, wird dabei automatisch berücksichtigt.

Die Ausgabe muß nicht unbedingt an den Drucker gehen, man kann das Ergebnis über den Menüpunkt »To ASCII File« auch auf Diskette und damit in ein anderes Programm übernehmen.

Softwood File II ist ein gut gemachtes Amiga-Programm. Leider leidet darunter an einigen Stellen die Leistungsfähigkeit. Die bedienerfreundliche Verwendung von Intuition mit seinen Windows und Requests ist eine einfache Angelegenheit. Schade nur, daß die Maus in diesem Programm etwas überstrapaziert wird. Wenn man nicht ganz so oft zwischen Tastatur und Maus wechseln müßte, ginge die Arbeit zügiger voran. Um sehr große Datenmengen schnell zu durchforsten, ist Softwood File II in dieser Version noch nicht optimal geeignet, aber für den Hausgebrauch ist das Programm völlig ausreichend und glänzt durch seine übersichtliche Handhabung, bei deren Realisierung manchmal aber die Arbeitsgeschwindigkeit nur an zweiter Stelle stand.

Gut ist auch, daß man für die Arbeit nicht die Originaldiskette verwenden muß. Man kann eine Arbeitskopie anfertigen und das Original im Schrank lassen. Bei Programmen, die aus Amerika importiert werden, wäre eine defekte Diskette sehr unangenehm.

Superbase auf dem Amiga

Superbase ist vielen Datenbank-Anwendern sicher schon vom C 64 her ein Begriff. Die C 64-Version ist für einen so kleinen Computer überaus leistungsfähig. Auch die Amiga-Version 1.0, die in Deutschland zusammen mit einem deutschen Handbuch vertrieben wird, ist schon im ersten Wurf recht gut gelungen. Sehr positiv ist, daß man sich die Mühe gemacht hat, das Programm, soweit es sinnvoll war, ins Deutsche zu übersetzen. Bis auf ein paar Unsauberkeiten erleichtert das die Bedienung ungemein.

Das Programm wird auf einer Diskette zusammen mit einem Kopierschutzstecker für den Joystick-Port 2 vertrieben. Deshalb ist auch kein Diskettenkopierschutz nötig, und man kann eine Arbeitsdiskette verwenden und wie bei Softwood File II das Original in den Schrank legen.

Superbase selbst ist recht kompakt programmiert, denn nach dem Start sind im Amiga

Die Amiga-Bibliothek



M. Breuer
Das Amiga-Handbuch
1986, 461 Seiten.
Kennenlernen und Anwenden der neuen Computertechnologie: Workbanch, Anwendersoftware, CLI, Grafik- und Soundmöglichkeiten, Mausbedienung.
Best.-Nr. 90228
ISBN 3-89090-228-6
DM 49,-/sFr 45,10/öS 382,20

M. Breuer
DELUXE Grafik mit dem Amiga
1987, 370 Seiten.
Schrittweise Einführung anhand überschaubarer Beispiele und Anwendung der wichtigsten Befehle. Datenaustausch zwischen den Programmen. Tips und Tricks für jeden Anwender.
Best.-Nr. 90412
ISBN 3-89090-412-2
DM 49,–/sFr 45,10/öS 382,20



W. Hilf/A. Nausch
CPU M68000 Teil 1: Grundlagen und Architektur
1984, 568 Seiten.
Ausbildungs- und Entwicklungstext mit allen notwendigen Informationen.
Bestell-Nr. 80316
ISBN 3-921803-16-0
DM 79,–/sFr 72,70/öS 616,20





D. Lien
Amiga-Programmier-Praxis mit MS BASIC
1986, 400 Seiten.
MS BASIC unter der komfortablen Maus/Fenster-Oberfläche des Amiga. BASIC in Musteranwendungen, Strings, Mathematik, Grafiken und vieles mehr.
Best.-Nr. 80369
ISBN 3-921803-69-1
DM 59,–/sFr 54,30/öS 460,20

M. Kohlen
Grafik auf dem Amiga
2. Quartal 1987, ca. 250 Seiten
Kennenlernen der fantastischen Möglichkeiten des Amiga-Computers. Ein Programmierkurs für Anfänger und Fortgeschrittene mit vielen Beispielen.
Best.-Nr. 90236
ISBN 3-89090-236-7
DM 49,-/sFr 45,10/öS 382,20

W. Hilf/A. Nausch
M68000-Familie Teil 2: Anwendung und 68000-Bausteine
1985, 400 Seiten
In vielen Programmierbeispielen liefert dieser Band die Praxis der in Teil 1 vermittelten Theorie.
Best.-Nr. 80330
ISBN 3-921803-30-6
DM 69,–/sFr 63,50/öS 538,20

Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser.

Irrtümer und Änderungen vorbehalten.



Zeitschriften · Bücher
Software · Schulung

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0.
SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656,
ÖSTERREICH: Rudolf Lechner&Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526, Ueberreuter Media Handels- und Verlagsges. mbH Großhandel, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0

Fragen Sie bei Ihrem Buchhändler nach unserem kostenlosen Gesamtverzeichnis mit über 200 aktuellen Computerbüchern und Software.

704124

noch etwa 160 KByte RAM frei, die zum Beispiel für den gleichzeitgen Betrieb des Sidecar völlig ausreichen.

Bevor man bei Superbase eine Dateistruktur definiert, sollte man sich Papier und Bleistift (oder das Notepad) zur Hand nehmen und sich die Reihenfolge der Felder, die man verwenden möchte, überlegen. Denn Sie können Felder für Text, Datum, für externe Dateien, Zahlen und Ergebnisse definieren. Das alles funktioniert sogar in vielen Variationen. Superbase überprüft auf Wunsch auch die Eingabe nach einer Formel, die Sie wählen, oder verlangt eine Feldeingabe zwingend. Es ist aber nicht gelungen, nachträglich die Reihenfolge der definierten Felder zu verändern. Sie können beim Arbeiten zwar einzelne Feldcharakteristiken ändern, aber eine neue Felddefinition an einer bestimmten Position einzufügen, scheint nicht möglich zu sein. Das ist an der Dateidefinition der einzige, aber doch gravierende Schwachpunkt, denn die Übersicht leidet erheblich, wenn jede neue Spalte oder Felddefinition nur rechts beziehungsweise unten angehängt werden kann.

Ist die Definition des Datenformats beendet, wählt man, welche Felder als Index für die Datei verwendet werden sollen. Das bedeutet, daß man, um eine Datei nach einem bestimmten Feld sortiert darzustellen, nicht wie bei Softwood File II neu sortieren muß. Man braucht nur eine andere Index-Datei anzuwählen, damit die Daten nach einem anderen Kriterium sortiert angezeigt werden. Superbase sortiert also den Datenbestand nicht jedesmal um, sondern benutzt nur eine Zeigerdatei auf die eigentlichen Daten. Das beschleunigt den Zugriff erheblich.

Mit Superbase kann man Datensätze auf unterschiedliche Weise in eine Datei einfügen. Neben der normalen Eingabe eines neuen Datensatzes über die Tastatur, kann man Daten aus einer ASCII-Datei importieren. Die Funktion »Import« filtert nach Bedarf die eingelesenen Daten und vergrößert die Datei um die neuen Datensätze.

Die Dateneingabe ist recht komfortabel gelöst. Alle neuen Daten können sofort ohne Probleme eingegeben und geändert werden. Ein Nachteil ist, daß man die Eingabe jedes einzelnen neuen Datensatzes bestätigen muß, damit er in die Datei übernommen wird. Wie bei Softwood File II ist also die flüssige Eingabe großer Datenmengen eher umständlich zu bewerkstelligen.

Superbase erlaubt auch die Verwendung von Daten im IFF-Format von Commodore. Das bedeutet, daß man zum Beispiel auch Bilder aus Dpaint in Datensätzen verwenden kann (Bild 3). Man muß nur entsprechende Felder mit dem Vermerk »Externe Datei« kennzeichnen. Bis jetzt kann Superbase aber »nur« Grafiken verarbeiten.

Um dem Benutzer die Verwaltungsarbeiten zu erleichtern, bietet Superbase im Menü »Prozeß« mehrere wichtige Funktionen. Mit »Löschen« werden aus der Datei alle Datensätze entfernt, die zum eingestellten Datenfilter passen. Mit der normalen Ausgabefunktion »Druck« wird alles, was vom Datenfilter durchgelassen wird, in Form einer Tabelle ausgedruckt. Eine besonders nützliche Funktion für viele Anwender ist sicher der Punkt »Etiketten«. Wer schon einmal viele Briefe verschickt hat, wird es sicher schätzen, daß Superbase für ihn die Etiketten beschriftet.

Auch Grafik als Datensatz

Auch für Ordnung in den Dateien gibt Superbase Hilfestellung. Man kann neue Index-Dateien genauso wie Dateien für Daten öffnen, alte löschen, die Ausgabe auf bestimmte Felder beschränken oder Abfragedefinitionen laden und speichern.

Superbase bietet zur Auswahl von Datensätzen eine ausgefeilte Filterfunktion (Bild 4). Sie erlaubt das Verknüpfen von mehreren Auswahlkriterien, die sich auf verschiedene Felder beziehen. Man kann Ausdrücke auch klammern. Besonders hervorzuheben ist die Auswahlfunktion »LIKE«. Mit ihr kann man Zeichenketten auf Ähnlichkeit vergleichen. Sie stellt eine stark erweiterte Version des Wildcard-Systems oder Pattern-Matching mit den Zeichen » * « und » ? « dar, was den Benutzern von MS-DOS und Commodore-Diskettenlaufwerken ein Begriff sein dürfte. Superbase erlaubt aber nicht nur einfach das Heraussuchen einzelner Datensätze auf dem Bildschirm. Man kann über die Menüfunktion »Abfrage« die Daten in fast unendlich vielen Variationen ausgeben. Superbase druckt die Daten in einer Formatierung, die Sie gewählt haben, summiert Zahlenkolonnen auf, errechnet Mittelwerte und filtert dabei die Ausgabe nach Bedarf. Ausgaben sind nicht nur auf dem Drucker möglich, man kann auch Disketten mit den Datensätzen beschreiben, das heißt, gefundene Datensätze exportieren. Damit Sie Ihre Abfragekriterien, die sehr umfangreich werden dürfen, nicht jedesmal neu eintippen müssen, sind sie auch auf Diskette speicherbar. Im Handbuch steht, die Abfragefunktion sei »das Herz von Superbase«. Das stimmt völlig. In einer Funktion so viele Variationsmöglichkeiten übersichtlich zu kombinieren, ist eine Kunst. Sich einen Überblick über große Datenmengen zu verschaffen, wird so zum Kinderspiel. Man braucht nur genügend Druckerpapier.

Superbase hat ein eigenes Menü »Einstellungen«, wo Sie Ihre Vorlieben und Vorgaben für die Arbeit wählen. Sie können die normale Ausgabe auf den Drucker umlenken oder den Bildschirm benutzen. Die Daten können spaltenweise als Tabelle, in geraffter Form als einzelner Datensatz oder als normales Eingabeformular gezeigt werden. Sollte Ihnen das Eingabeformular formal-ästhetisch nicht gefallen, verschieben Sie die einzelnen Felder mit der Maus. Das Formular wird mit der Datei gespeichert und muß also nicht immer neu eingestellt werden.

Inzwischen schon üblich, aber bei Superbase flexibler als normal, ist die Möglichkeit, das Format des Datums und von Zahlen auf individuelle oder landesspezifische Notwendigkeiten umzustellen. Dazu gehören verschiedene Währungssymbole genauso, wie Punkt oder Komma als Trennzeichen bei Zahlen. Das Datum, das Superbase in seinen Ausgaben benutzt, wird hier auch eingestellt.

Damit man nicht zwischen mehreren Utilities und Superbase hin- und herwechseln muß, bietet Superbase im Menü »System« wichtige Grundfunktionen, die das Programm wie auch die Umgebung im Computer beeinflussen. Man kann das aktuelle Inhaltsverzeichnis der Diskette abrufen oder ein neues anwählen. Wenn nötig, rufen Sie mit der Funktion »Status« einen Zustandsbericht über die aktuell gewählten Dateien oder über den Computer und dessen Speicherverhältnisse ab. Mit der Funktion »Textausgabe« listen Sie eine ASCII-Datei auf dem Bildschirm, wenn Sie kurz eine Zusatzinformation aus einem bestimmten Text benötigen. Superbase besitzt aber auch eine Hardcopy-Funktion für den Bildschirm, die in der deutschen Version den schönen Namen »Bildschirmkopie« trägt.

Für jeden etwas

Die wichtigste Funktion für die, die sehr viel mit Superbase arbeiten, ist in diesem Menü wohl »Reorganisation«. Eine Datei, die sehr stark bearbeitet und geändert wurde, wird damit neu aufgebaut. Der Zugriff auf die Daten erfolgt dann wieder schneller, da alle unnützen Informationen aus der Datei entfernt worden sind.

Superbase ist schon fast eine professionelle Datenbank, die auch für große Datenmengen benutzt werden kann. Das Programm nutzt Intuition genau im richtigen Maß. Es gibt keine unnötigen Spielereien, die zusätzlichen Speicherplatz benötigen. Superbase bietet für ein Programm in der Version 1.0 schon außergewöhnlich viele Möglichkeiten. Leider ist Superbase Amiga in dieser Version nicht programmierbar, obwohl im Handbuchanhang schon der komplette Befehlssatz und alle Fehlermeldungen der Superbase-Datenbanksprache abgedruckt sind. Sogar im Programmcode selbst sind diese Texttabellen und auch das entsprechende Menü schon vorhanden. Es wird nur nicht angezeigt. Die Firma Precision Software hätte sich vielleicht noch etwas Zeit nehmen sollen, um dies fehlerfrei zu integrieren. Wäre Superbase zusätzlich programmierbar, hätte man ein noch viel leistungsfähigeres Programm als ohnehin schon.

Eine leichte Schwachstelle ist auch das Handbuch, denn für einen Anfänger in Datenverwaltung ist es zu unübersichtlich aufgebaut. Ist man aber mit den Grundbegriffen vertraut, ist das Handbuch sehr aufschlußreich, da es viele Beispiele enthält.

Alles in allem ist Superbase eine sehr leistungsfähige Datenbank. (Heinz Wrobel/nj)

Info: Softwood File II: B. Soltau KG, Esplanade 39, 2000 Hamburg 36, Preis: 235 Mark
Superbase: Markt&Technik Verlag, Hans-Pinsel-Str. 2, 8013 Haar bei München, 298 Mark

Amiga als Kalkulationsgenie

Ist der Amiga nur eine Spiel- und Grafikmaschine? Oder eignet er sich auch für den kaufmännischen Einsatz? Mit den von uns getesteten Tabellenkalkulationen wird der Amiga zum ernsthaften Helfer für jedes Büro.

Eine der Hauptaufgaben moderner Computer ist die Unterstützung des Menschen bei Routinearbeiten, die noch dazu sehr viel Zeit in Anspruch nehmen. Während noch vor wenigen Jahren für diese Aufgaben Großrechner und Personal Computer im Einsatz waren, sind die Grenzen zwischen Heim- und Personal Computer in der Zwischenzeit so stark verwischt, daß es nur noch sehr schwer möglich ist, einen bestimmten Computer einer der beiden Kategorien zuzuordnen. Betrachtet man den Amiga mit seinen erstaunlichen Grafik- und Soundfähigkeiten, wird die Abgrenzung noch schwieriger. Auf der einen Seite gibt es faszinierende Spiele, was den Amiga oft zum Spielcomputer verdammt. Auf der anderen Seite gibt es hervorragende Grafik-, CAD- und Musikprogramme, die den Amiga voll ausnützen. Wie aber sieht es im Bereich der kaufmännischen Anwendung aus? Eines der wichtigsten Werkzeuge für den Kaufmann ist die Tabellenkalkulation. Hier können Zahlen übersichtlich dargestellt, Berechnungen durchgeführt und Werte grafisch veranschaulicht werden. Bedenkt man die oft großen Mengen an Daten die in diesem Bereich anfallen, stößt man hier mit gewohnter Handarbeit sehr schnell an die Grenzen menschlichen Leistungsvermögens. Stellen Sie sich vor, Sie müßten über 1000 Zahlen in ein tabellenartiges Arbeitsblatt eintragen und dann noch alle nur erdenklichen Quer- und Gesamtsummen und die zugehörigen Verhältnisse in Prozent berechnen. Wenn sich dann auch noch ein kleiner Fehler einschleicht, ist die ganze Arbeit, die bei dieser Arbeitsweise Wochen in Anspruch nimmt, umsonst getan.

Komfortable Eingabe

Sehr schnell kam man hier auf die Idee, die elektronischen Helfer für solche Aufgaben einzuspannen. Die ersten Tabellenkalkulationen auf dem Computer entstanden. In der heutigen Zeit der Personal Computer ist das Angebot an derartigen Programmen reichhaltig. Da für solche Aufgaben Geschwindigkeit und Grafikfähigkeit gefragt sind, ist der Amiga dafür prädestiniert. Wir haben uns einige Produkte für den Amiga angesehen, gespannt darauf, was uns hier an Geschwindigkeit und Grafik erwartet. Zum Testen hatten wir drei Tabellenkalkulationen vorliegen: Analyze 2.0, VIP Professional und Logistix.

Bevor wir uns die Leistungsfähigkeit in Sachen Grafik näher betrachten, die durchaus unterschiedlich ist, sehen wir uns diejenigen Bestandteile näher an, die alle Programme gemeinsam haben. Dazu muß man wissen, daß das Programm, wie auch der Mensch, mit Arbeitsblättern, im Fachjargon »Worksheets« genannt, arbeitet. Diese Arbeitsblätter sind wiederum in Zellen unterteilt. Eine Zelle stellt eine einzelne Informationseinheit dar, die sich durch Befehle gezielt ansprechen läßt. Eine Zelle läßt sich durch das im Arbeitsblatt verwendete Koordinatensystem ansprechen (Bild 1). Die Zellbezeichnung, gegeben durch die Koordinaten, wird hier als Variable für alle Operationen verwendet. An Operationen haben alle drei Programme einiges zu bieten. Da wäre als erstes die Dateneingabe zu nennen. Eine Zelle wird in einer Eingabezeile über dem Arbeitsblatt editiert. Das heißt hier werden die Werte erfaßt und gegebenenfalls verändert. Die Werte können hier nach verschiedenen Kriterien eingegeben werden. Zum einen sind da Texte, die dokumentierend in die Tabelle eingefügt werden. Diese werden automatisch als solche erkannt und bei Berechnungen übergangen. Am wichtigsten sind allerdings die numerischen Eingaben. Diese werden ebenfalls vom Programm erkannt und zu späteren Berechnungen herangezogen. Nun stellt sich natürlich die berechtigte Frage, wie man denn nun lange Zahlenreihen verarbeiten kann. Hierzu bieten alle drei Kandidaten eine leistungsfähige Programmiersprache an.

Die Sprache hält sich größtenteils an die aus der PC-Welt bekannte Tabellenkalkulation Lotus 1-2-3. Interessant ist vor allem die Verwaltung von Werten, die mit den eingebauten Befehlen ermittelt werden. Haben Sie beispielsweise eine Zahlenreihe aufaddiert und das Ergebnis in eine andere Zelle ausgelagert, finden Sie beim Bearbeiten dieser Zelle nicht die entsprechende Zahl in der Editierzeile, sondern die für die Berechnung angegebene Formel. Theoretisch können Sie vor der Eingabe der Werte bereits die Ergebnisfelder festlegen, da diese automatisch geändert werden. Wie aber sieht nun eine solche Formel aus und wie vor allem wird sie angewendet?

Großes Vorbild

Das betrachten wir jetzt an einem kleinen Beispiel. Nehmen wir an, Sie hätten in den Zellen »A1« bis »A5« Werte stehen, deren Summe in Zelle



Bild 1. Aufbau eines Arbeitsblattes am Beispiel von Analyze

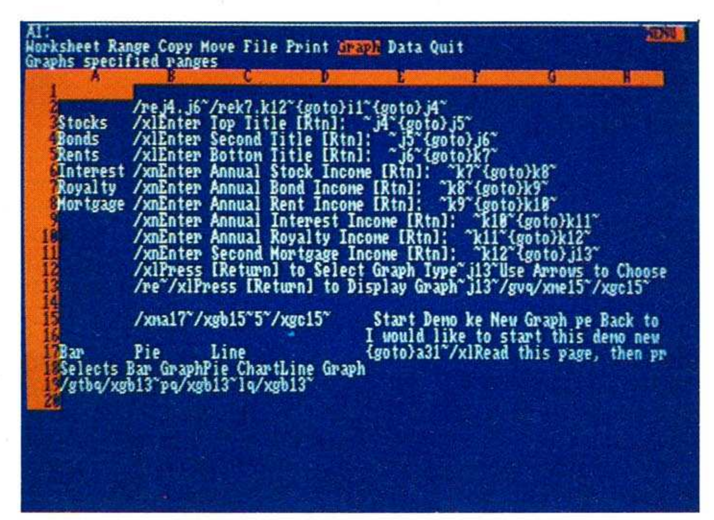


Bild 2. Das Arbeitsblatt von VIP mit Programmanweisungen

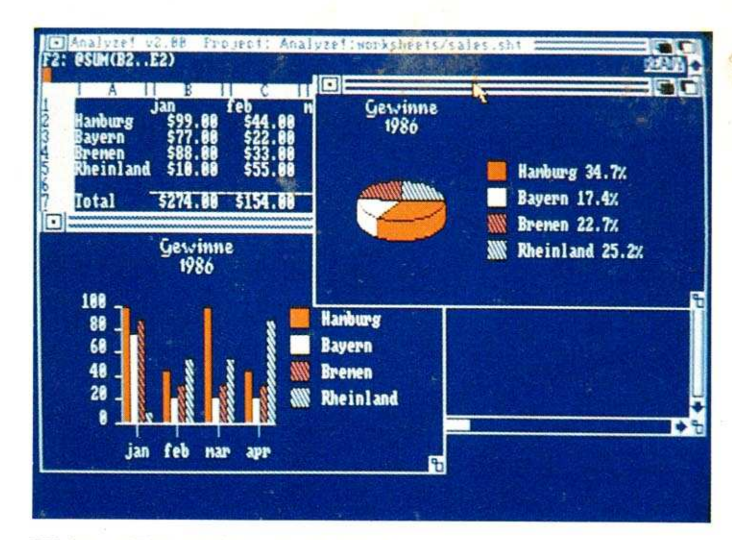


Bild 3. 4-Farben-Grafik mit Analyze

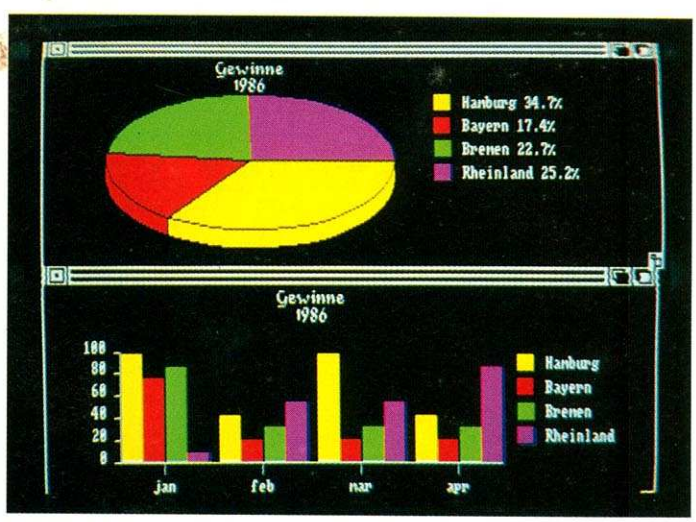


Bild 4. 8-Farben-Grafik, erstellt mit Analyze

»A6« stehen soll. Dazu bieten die Programme den Befehl »SUM« an. Editieren Sie Zelle »A6«, geben Sie in diesem Fall keine Zahl ein, sondern eine Formel, die die genannten Zellen aufaddiert:

```
SUM(A1..A5)
```

Auf diese Art und Weise können auch alle anderen Befehle in einer Art Direktmodus verwendet werden. Was aber nützt die beste Sprache, wenn man nicht auch Programme damit schreiben kann? Natürlich bieten die Tabellenkalkulationen diese Möglichkeit. Dazu werden die Befehle und Parameter als Text in die einzelnen Zellen geschrieben. Dabei können Schleifen gebaut, Abfragen getätigt und Sprungziele gesetzt werden. Der Tabellenkalkulation muß dann nur noch mitgeteilt werden, daß es sich beim Inhalt eines bestimmten Bereichs um ein Programm handelt, damit hier nicht fälschlicherweise normaler Text verarbeitet wird. Dann aber lassen sich große Arbeitsblätter kalkulieren, die auf Knopfdruck nach bestimmten Kriterien sortiert oder durchgerechnet werden.

Neben den umfangreichen Befehlen zum Berechnen von mehreren Werten gibt es noch einige, die sich mit der Gestaltung des Arbeitsblattes selbst befassen. Damit können dann einzelne Zellen oder ganze Zellbereiche innerhalb des Arbeitsblattes verschoben oder kopiert werden. Ebenfalls wichtig ist das Zahlenformat, das für die verschiedenen Werte gültig ist. Hier können Sie nicht nur die Anzahl der Nachkommastellen einstellen, sondern sich auch für eine Anzeige mit vorangestellter Währung entscheiden. Prozentdarstellungen sind ebenfalls kein Problem.

Nachdem wir uns nun näher mit der Programmierung beschäftigt haben, betrachten wir uns nun den Eingabekomfort. Hier weisen die drei Testkandidaten gravierende Unterschiede auf. Bei VIP kann der Anwender den Zellcursor nur mit den Cursor-Tasten steuern, ein Positionieren mit der Maus ist nicht möglich. Dieselbe Einschränkung zeigt sich auch bei Logistix. Bei Logistix handelt es sich um eine Umsetzung bewährter MS-DOS-Software, so daß hier die Maus nicht in besonderem Maße unterstützt wird. Dafür bietet Logistix noch einige besondere Möglichkeiten, auf die wir später noch genauer eingehen werden.

Am besten gelöst ist das Problem der Eingabe zweifellos bei Analyze. Hier kann der Zellpointer beliebig mit Hilfe der Maus positioniert werden. Hat man Arbeitsblätter, deren Grenzen über den Bildschirmrand hinausgehen, kann man den Arbeitsblattausschnitt mit Hilfe von Rollbalken am unteren und rechten Bildschirmrand beliebig in alle Richtungen scrollen. So erreicht man mühelos alle beliebigen Punkte auf dem Arbeitsblatt. Bei den beiden anderen Testkandidaten erreicht man dies wie bereits erwähnt nur mit den Cursortasten oder einem speziellen GOTO-Befehl im Direktmodus. Bei Analyze dient die Maus auch zum Markieren ganzer Bereiche, die dann verschoben, kopiert oder mit Befehlen bearbeitet werden. Bei Logistix und VIP müssen hier die genauen Zellkoordinaten angegeben werden. Bei der Eingabe also ein klarer Vorteil für den Analyze-Anwender. Allerdings sind die Befehle von Analyze nicht komplett kompatibel zu Lotus 1-2-3. Hier stellt sich allerdings die Frage, was eine Lotus-Kompatibilität beim Amiga bringen soll, da Lotus für den Amiga noch nicht erhältlich ist.

Nicht immer mit der Maus

Bestimmte Aktionen lassen sich, wie bereits erwähnt, im Direktmodus durchführen. Auch hier gibt es wieder große Unterschiede zwischen den drei Testkandidaten. In den Befehlsmodus gelangt man bei allen drei Programmen mit Hilfe der </>-Taste. Dann erscheint ein, je nach Programm mehr oder weniger umfangreiches Menü, aus dem durch Drücken der Anfangsbuchstaben ausgewählt wird. Hier werden alle Funktionen zum Laden und Speichern von Arbeitsblättern oder zum Aufbau einer Grafik gestartet.

Neben dem Aufruf über die </>-Taste bietet Analyze hier noch sehr komfortable Pull-Down-Menüs an, die die Arbeit vor allem für den Anfänger stark erleichtern. Kopieren oder Verschieben von Bereichen erfolgt ohne einen einzigen Tastendruck nur mit Hilfe der Maus. Allerdings muß an dieser Stelle erwähnt werden, daß Analyze speziell für den Amiga entwickelt wurde.

Als nächstes wollen wir uns einen der wichtigsten Bestandteile einer Tabellenkalkulation, die Programmiersprache, näher ansehen. Im Großen und Ganzen stimmen hier alle drei Kandidaten überein, was die Tabellenkalkulation betrifft. VIP und Analyze verwenden englische Befehle zu Berechnungen und grafischen Darstellungen. Bei Logistix, das soweit als möglich eingedeutscht ist, findet man ein Gemisch aus englischen und deutschen Worten. Neben den vier Grundrechenarten stehen hilfreiche Winkelfunktionen zur Verfügung. So können die Tabellenkalkulationen auch zu technischen Berechnungen herangezogen werden.

Überzeugende Grafik

Daneben lassen sich über diverse logische Funktionen Programmfehler abfangen, was eine komfortable Programmierung erlaubt. Natürlich steht dazu auch eine entsprechende IF-Abfrage zur Verfügung. So ganz wie im gewohnten Basic geht es allerdings bei der Programmierung einer Tabellenkalkulation nicht gerade zu. Betrachtet man ein derartiges Programm (Bild 2) näher, fällt als erstes die Abstraktheit auf. Hinter den Befehlen verbirgt sich aber nichts anderes als die Anfangsbuchstaben aus den Menüs der Tabellenkalkulation.

Bisher haben wir uns nur mit grauen Zahlen und was man mit diesen anstellen kann, beschäftigt. Für das Auge aber sind endlose Zahlenkolonnen relativ nichtssagend. Vor allem der Amiga ist hier prädestiniert für grafische Ausdrucksformen. Nicht zuletzt aus diesem Grund kann mit allen drei getesteten Programmen eine Grafikdarstellung verwirklicht werden. Um Grafiken darstellen zu können, müssen erst bestimmte Bereiche des Arbeitsblattes markiert werden. Daneben können die Kommentare als Beschriftung für die einzelnen

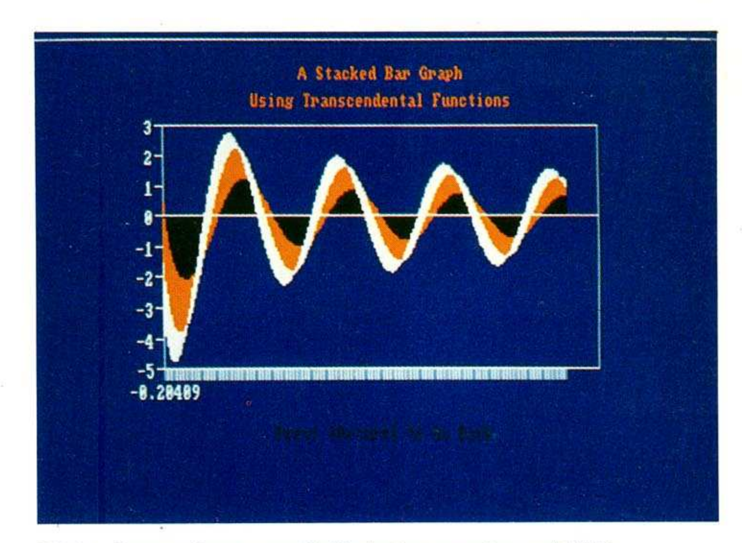


Bild 5. Darstellung von Kalkulationswerten mit VIP

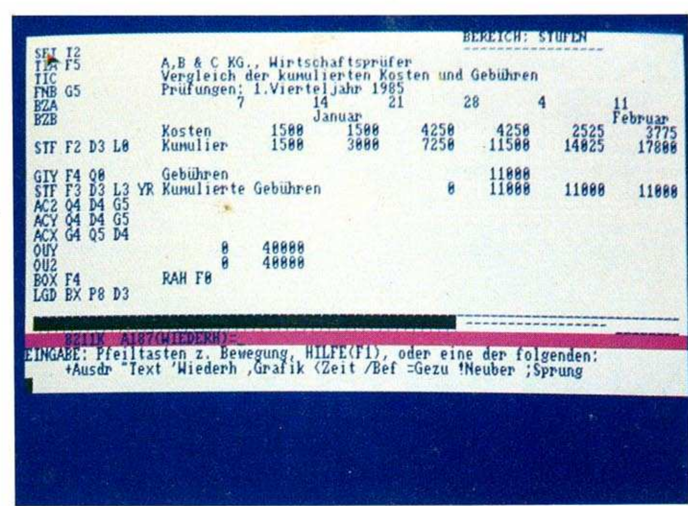


Bild 6. So präsentieren sich die Werte des Terminplaners bei Logistix

Achsen verwendet werden. Analyze bietet hier wieder umfangreichere Möglichkeiten. Es können nicht nur acht verschiedene Grafikformen angewählt, sondern auch unterschiedliche Farbauflösungen verwendet werden. Dabei lassen sich mehrere Grafiken gleichzeitig öffnen. Prinzipiell werden die Grafiken in frei verschiebbaren Fenstern angezeigt, deren Größe sich nach Belieben ändern läßt. Arbeitet man mit vier Farben, werden die Fenster mit der Grafik über das Arbeitsblatt gelegt, können aber jederzeit ausgeblendet werden. Ist eine Grafik aufgebaut und es werden dafür relevante Werte geändert, erfolgt nicht nur automatisch die Neuberechnung des Arbeitsblattes, sondern auch die Grafik wird sofort angepaßt. Arbeitet man mit achtfarbiger Grafik, werden die Fenster voll über das Arbeitsblatt gelegt, so daß ein Bearbeiten der Werte nur über das Ausblenden der Grafik möglich ist. Ein Beispiel für beide Arten sehen Sie in Bild 3 und 4. Als kleine Zugabe lassen sich die Fonts (Zeichensätze), mit denen die Beschriftung der Grafiken erfolgt, ändern. Hier finden die Fonts Verwendung, die auf der Amiga-Workbench vorhanden sind. Sie sehen, daß die Grafiken optisch und faktisch sehr repräsentativ und aussagekräftig sind. Bis auf die Eingabe der Grafik-Titel werden alle für den Aufbau der Grafik notwendigen Schritte mit der Maus durchgeführt. Für ein angenehmes Arbeiten ist also durch die Ausnutzung der Fähigkeiten des Amiga gesorgt. Alternativ kann hier natürlich auch über die entsprechenden Tastaturbefehle vorgegangen werden. Bei Logistix und VIP

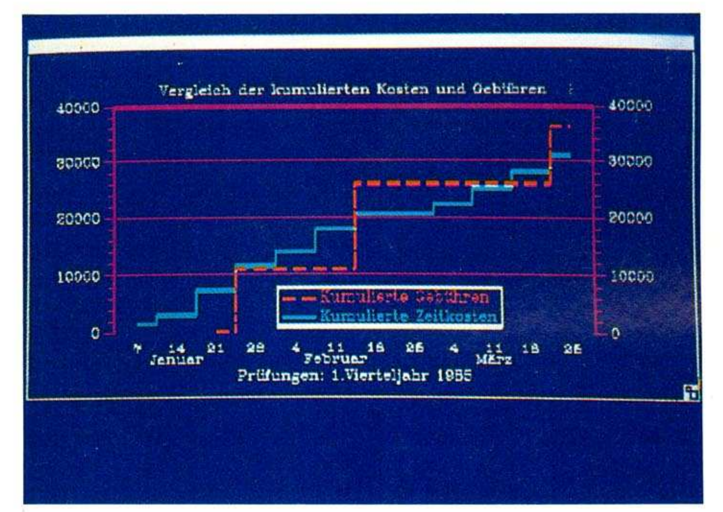


Bild 7. Grafische Darstellung eines Terminplans, der mit Logistix erstellt wurde

bleibt überhaupt nur diese Möglichkeit. Außerdem stehen bei VIP nur fünf verschiedene Formen für die Grafik zur Verfügung. Eine davon sehen Sie in Bild 5. Daneben wird mit VIP ein zweites Programm ausgeliefert, das nur für den Ausdruck der Grafiken zuständig ist. Eine praktische Einrichtung, wenn man nur mal eben eine Statistik ausdrucken will. Logistix bietet eine Auswahl aus neun verschiedenen Grafiktypen. Dabei können auch Terminplanungen grafisch dargestellt werden (Bilder 6 und 7). Die gravierendsten Unterschiede zwischen den drei Testkandidaten finden Sie der besseren Übersicht wegen in Tabelle 1 aufgelistet. Alle Gemeinsamkeiten unter den Programmen sind nun, was die Tabellenkalkulation selbst angeht, aufgeführt.

Während Analyze neben der Tabellenkalkulation keine weiteren Funktionen eingebaut hat, kann man mit Logistix und VIP noch einiges mehr anstellen, als nur Arbeitsblätter berechnen. Beide Programme haben gleichzeitig eine Art Datenbank eingebaut, wie das auch beim Vorbild, Lotus 1-2-3, der Fall ist. Allerdings gibt es dafür keine speziellen Masken beziehungsweise Bildschirmaufbauten, es wird das Arbeitsblatt der Tabellenkalkulation verwendet.

Integrierte Datenbank

Die Datenbank ist hier vor allem für Statistiken, Analysen und finanzielle Planspiele geeignet. Kennzeichnend für eine Datenbank ist die Tatsache, daß die einzelnen Zellen wie die Felder eines Datensatzes behandelt werden. Eine Reihe auf dem Arbeitsblatt stellt jeweils einen kompletten Datensatz dar. Dabei müssen die Zellen der ersten Reihe (Zellen A1, B1, C1 usw.) immer die Feldnamen beinhalten, unter denen die einzelnen Datenfelder angesprochen werden. Zur Bearbeitung der Datenbanken bietet VIP noch einige Zusatzbefehle, die die Handhabung erleichtern. Da sich auch die Informationen aus einer Datenbank grafisch darstellen lassen, steht umfangreichen Statistiken, beispielsweise über Bevölkerungsdichten, nichts im Wege. Zwar kann man den Aufbau einer Datenbank mit Hilfe eines Arbeitsblattes nicht gerade zu den komfortabelsten Methoden zählen, doch lassen sich trotzdem erstaunliche Ergebnisse erzielen. Damit wäre auch die Besprechung von VIP beendet, da keine weiteren Anwendungen mehr zur Verfügung stehen. Wäre noch anzumerken, daß mit einem Amiga, der über einen Speicher von 512 KByte verfügt, lediglich 29 KByte für das Arbeitsblatt vorhanden sind. Das reicht teilweise nicht einmal für die mitgelieferten Demos aus, so daß man bei Anwendung von VIP um eine Speichererweiterung nicht herumkommt. VIP kostet mit englischem Handbuch 499 Mark, ein Preis der für den Heimanwender etwas zu hoch bemessen sein dürfte. Für kommerzielle Anwender hat VIP zu diesem Preis aber einiges zu bieten.

Sehen wir uns nun die Besonderheiten des wohl mächtigsten Testkandidaten an: Logistix. Der Hersteller bezeichnet Logistix nicht nur als Tabellenkalkulation, sondern als integriertes Softwarepaket. Darunter versteht man Programme, die Funktionen in sich vereinigen, zu deren Ausführung normalerweise verschiedene Programme notwendig sind. Die modernsten integrierten Pakete aus der PC-Welt beinhalten Textverarbeitung, Tabellenkalkulation und Datenbank,

sowie diverse Grafikfunktionen. Ganz so umfangreich ist Logistix zwar nicht, hat aber dennoch reichhaltige Funktionen integriert, die die Bezeichnung »integriertes Paket« rechtfertigen.

Außer Textverarbeitung bietet Logistix Tabellenkalkulation, Datenbank und Terminplanung. Gerade der letzte Punkt ist für alle interessant, die umfangreiche Terminkalender zu verwalten haben. Die Terminplanung erleichtert neben einer umfangreichen Übersicht über die eigenen Termine auch die Aufteilung der Arbeit. Letzteres trifft wohl mehr auf Vorgesetzte und Arbeitsvorbereiter zu, die zeitlich begrenzte Fertigungsaufträge in genaue Zeitvorgaben und zeitlich abgegrenzte Fertigungsschritte aufteilen müssen. Bevor wir nun die Funktion des Terminplaners näher unter die Lupe nehmen, sei noch auf eine Besonderheit von Logistix aufmerksam gemacht. Alle Kommentare, wie Fehlermeldungen oder Hilfstexte sind ebenfalls wie das umfangreiche Handbuch in Deutsch gehalten. So hat der Anwender keine allzu großen Schwierigkeiten, sich in das Paket einzuarbeiten. Einige Funktionen sind jedoch etwas unzureichend beschrieben, so daß man an einer gewissen »Probierphase« nicht vorbeikommt. Daneben wird Logistix mit einem »Dongle« ausgeliefert. Dabei handelt es sich um einen kleinen Stecker für den zweiten Joystick-Anschluß des Amiga. Dieses Dongle beherbergt eine kleine Schaltung, die von Logistix abgefragt wird. Ohne das Dongle kann das Programm nicht gestartet werden. Ein beinahe perfekter Kopierschutz. Außerdem können durch diese Methode jederzeit Sicherheitskopien der Disketten angefertigt werden. Leider hat die Sache mit dem Dongle einen entscheidenden Nachteil: Mit dem neuen Amiga 500 arbeitet Logistix (in der getesteten Version noch) nicht zusammen. Beim Versuch das Programm zu starten, erscheint lediglich ein Window mit der lapidaren Meldung, das Dongle wäre nicht vorhanden. Und das obwohl dieser ordnungsgemäß angebracht wurde. Die Software-Entwickler werden sich hier noch Gedanken machen müssen, wie man zu einer allgemein gültigen Version kommen kann. Ein Problem, das nicht nur für Logistix gilt.

Für eine Terminplanung ist natürlich immer ein Kalender notwendig. Außerdem muß berücksichtigt werden, in welchen Zeiteinheiten (Monate, Tage, Stunden) die Termine aufgebaut sind. Um all diese Anforderungen zu erfüllen gibt es bei Logistix ein eigenes Untermenü mit diversen Kalenderfunktionen. Hier braucht nur die gewünschte Zeiteinheit eingestellt und das Tagesdatum eingegeben werden, und schon steht ein kompletter Kalender über mehrere Monate zur Verfügung. Editiert man den Kalender, stellt man überrascht fest, daß alle Wochentage richtig gesetzt und die Wochenenden farbig hervorgehoben sind. Unter dem Kalender findet sich dann noch eine Zeitleiste, deren Aufteilung Arbeitspausen berücksichtigt. Den Kalender kann man dann per Befehl in das Arbeitsblatt übertragen und die verschiedenen Arbeiten für einen Terminauftrag erfassen.

Grafik und Zeit

So entsteht sehr schnell ein netzplanartiges Gebilde, das auf einen Blick alle Informationen über den zeitlichen Ablauf eines Auftrags wiedergibt. Wer sich etwas mit der komplexen Theorie der Netzplan-Technik, mit der sich im kaufmännischen Bereich zeitliche Abläufe darstellen lassen, auskennt, weiß, daß mit diesem umfangreichen Werkzeug ganze Projekte geplant, beziehungsweise organisiert werden. Daneben garantiert diese Art der Terminorganisation eine beinahe optimale Überwachung der Terminsituation. Die Anwendungsbereiche reichen hier von der privaten Zeiteinteilung bis hin zur kaufmännischen Projektverwaltung. Eine Funktion also, die für den Privatmann oder Lehrer genauso nützlich ist, wie für den Geschäftsmann und Manager.

Für kommerzielle Anwendung bestens geeignet, so könnte man den Amiga, nachdem man die Leistungsfähigkeit der Tabellenkalkulationen Revue passieren läßt, bezeichnen. Unsere drei Testkandidaten halten sich an den von den PCs gesetzten Standard und bieten darüber hinaus verbesserte Grafikfunktionen, sowie höheren Bedienungskomfort und auch Kompatibilität untereinander. Durch die bei allen Programmen vorhandene »Lotus-Kompatibilität« ist gewährleistet, daß Daten ohne Probleme untereinander ausgetauscht werden können. Wenn man bedenkt, daß auch im Bereich der kommerziellen Anwendung die grafische Darstellung trockener Daten immer wichtiger wird, bleibt abschließend zu sagen, daß der Amiga für derartige Aufgaben bestens geeignet ist. (rf)

Die wichtigsten Informationen auf einen Blick

	Analyze	Logistix	VIP
Anzahl Zellen	256 x 8156	2048 x 1024	256 x 8156
Steuerung	Maus	Tastatur	Tastatur
Funktionen	Kalkulation	Kalkulation Datenbank Terminplanung	Kalkulation Datenbank
Preis	349 Mark	499 Mark	499 Mark

Info: Softwareland, Franklinstr. 27, CH-8050 Zürich, vertreibt alle drei Programme
Computer Technik Kieckbusch GmbH, Am Seeufer 11 + 22, 5412 Ransbach, VIP Professional
Interplan GmbH, Nymphenburger Str. 134, 8000 München 19, vertreibt alle drei Programme

Malen leichtgemacht

AMIGA TEST Beim Wort »Amiga« denkt man natürlich sofort an hervorragende Grafik. Mit Aegis Images haben Sie auf einfache Weise Zugriff darauf.

Da Aegis Images als Zugabe zum Aegis-Animator (Test in dieser Ausgabe) geliefert wird, mag wohl so mancher ratlose Software-Käufer denken: »Das ist doch wieder so ein Mini-Extra wie das Malprogramm Graphicraft. Doch weit gefehlt. Natürlich besitzt Images alle wichtigen Grundfunktionen wie das Zeichnen einer Linie, das Füllen einer geschlossenen Kurve, das Entwerfen von Kreisen und Ellipsen und das »freihändige« Zeichnen mit der Maus. Doch schon bei diesen Funktionen wird es interessant: Unter dem Schlagwort »Shape« verstecken sich einige interessante Möglichkeiten, den Grundfunktionen ein wenig auf die Sprünge zu helfen. Es ist zum Beispiel bei allen geometrischen Eckformen (wozu Figuren mit 2 bis 19 Ecken gehören) möglich, beim Zeichnen einen oder zwei Eckpunkte »festzuhalten«. Die anderen Eckpunkte lassen sich dann mit der Maus auf dem Bildschirm bewegen. So lassen sich reizvolle Effekte erzielen. Runde Körper wie Kreis oder Ellipse können ebenso auf diese Weise bearbeitet werden, wobei das Programm deren Mittelpunkte am Bildschirm sozusagen »festgeklemmt« und die Radien verändert werden. Am wichtigsten ist bei einem Grafikprogramm auf dem Amiga natürlich die Farbeinstellung, denn diesen Computer zeichnet ja seine Farbenpracht aus. Images arbeitet in der niedrigen und mittleren Auflösung des Amiga, ein Bild kann also bis 32 beziehungsweise 16 verschiedene Farben haben, die aus der reichhaltigen Farbpalette ausgesucht werden können. Der hochauflösende Modus des Amiga wird nicht unterstützt, was aber wegen des dabei auftretenden Flimmerns sowieso nicht zu längerem Arbeiten verleitet. Es sei denn, man würde eine Anti-Flicker-Blende benutzen.

Wer beim Malen oft die Farbe wechselt, wird das »Fast Menu« schätzen lernen (Bild 1). Das ist ein kleines Fenster auf dem Bildschirm, das die Palette zur Farbauswahl sowie die meistgebrauchten Pinsel- und Zeichenarten enthält und mit der Maus beliebig umhergeschoben werden kann (Bild 2). So erspart man sich die im Vergleich dazu umständlichen Pull-Down-Menüs. Wer es nicht braucht, kann das Fast-Menu natürlich auch ausschalten. Es ist auch möglich, die wichtigsten Funktionen des Programms per Tastatur anzuwählen, was versierten Anwendern viele Mausklicks erspart.



Bild 1. Für Einsteiger gibt es das »Fast-Menu«



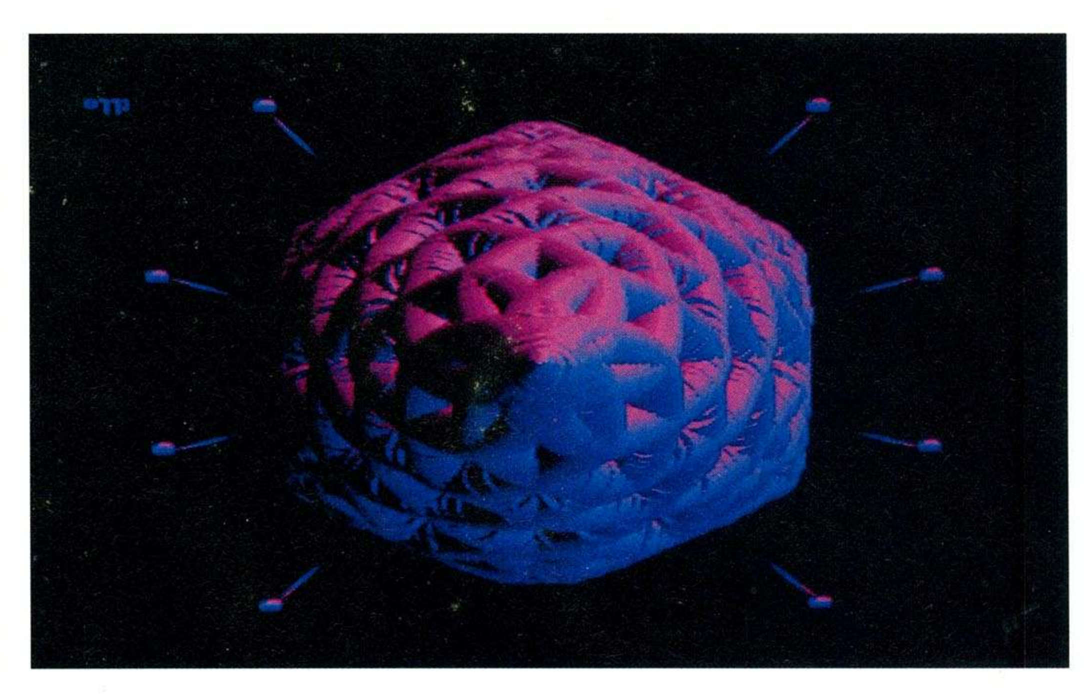
Bild 2. Mit »Images« lassen sich tolle Grafiken erstellen

Doch mit den Farbmenüs lassen sich noch eine Menge andere Dinge auf den Bildschirm bringen. Durch geschickte Rechentricks erzeugt das Programm einen Transparenz-Effekt, der so wirkt, als ob man durch farbigen Rauch sein Bild sieht. Der »Under«-Modus verhindert das Übermalen von bestimmten Farben der Zeichnung. Mit »Wipe« (Wischen) und »Smear« (Schmieren) lassen sich aquarellartige, ineinander verschwimmende Farbübergänge herstellen. All diese Menüpunkte erzeugen reizvolle Farbeffekte — wenn man sich nicht an dem in diesen Punkten schon fast unerträglich langsamen Arbeitstempo stört, wobei der Pinsel manchmal nur noch zwei Schritte pro Sekunde auf dem Bildschirm vorankommt. Daß es auch schneller gehen kann, beweist Deluxe-Paint: Hier benötigen dieselben Effekte nur eine kaum merkbare Verzögerung. Neben den verschiedenen Pinseln wie Rechteck oder Kreis stehen auch selbstdefinierbare Muster (Patterns) zur Verfügung. Das sind 8 x 8 Pixel große Musterbausteine, mit denen ganz normal gemalt werden kann. Es erscheint eben statt einer geschlossenen Fläche ein farbiges Muster. Um neue Muster zu entwerfen, kann man entweder einen Editor benutzen oder kleine Teile einer vorhandenen Zeichnung verwenden. Wer aber mit größeren Teilen der Zeichnung arbeiten will, ist mit »Frame« (Rahmen) besser bedient. Dabei kann ein beliebiger Ausschnitt aus der Zeichnung markiert und dann vergrößert, verzerrt, verkleinert, gedreht und auf Diskette gespeichert werden. Bei jedem Grafikprogramm gibt es eine Füll-Funktion, die umschlossene Flächen ausfüllt. Aegis hat diese Funktion für Images stark erweitert: Es ist nun möglich, geschlossene Flächen mit selbst auswählbaren Farbabstufungen zu füllen. Besonders interessant ist dabei die Möglichkeit, die Farben beliebig stark ineinanderfließen zu lassen. Von harten Farbrändern bis zu einer völligen Verschmelzung der Farben ist alles machbar.

Unter den sonstigen Menüpunkten von Aegis Images fällt die gut gelungene Sprühdosen-Funktion auf, bei der sich Sprühstrahl und -intensität beliebig ändern lassen. Ein zweiter Zeichenbildschirm wurde ebensowenig vergessen.

Alles in allem wäre Aegis Images auch als einzelnes Programm (ohne Aegis Animator) seinen Preis wert, wenn man sich durch die teilweise recht geringe Arbeitsgeschwindigkeit nicht abschrekken läßt. Daß es dem Animator beiliegt, kommt dem Anwender natürlich entgegen. Für das gut aufgemachte englische Handbuch sollte der Anwender zumindest Grundkenntnisse der englischen Sprache mitbringen, sonst dürfte ihm das Ausreizen des Programms schwerfallen. Als sehr gut bleibt zu vermerken, daß Aegis im Interesse der ehrlichen Anwender auf einen Kopierschutz verzichtet hat. So wird ein Verlust der Originaldiskette nicht zu einer mittleren Katastrophe.

(Andreas Lietz/jk)



Deluxe-Paint II — das Grafikwunder

AMIGA TEST Der Amiga ist in Sachen Grafik derzeit einer der leistungsfähigsten Computer. Doch die Hardware ist nur so gut wie das Programm, das auf sie zugreift. Wir testen für Sie das Zeichenprogramm, das alle Tricks des Amiga ausnutzt: Deluxe-Paint II.

Ein sehr beliebtes Thema bei Computerfreunden ist alles, was etwas mit Grafik zu tun hat. Wir stellen Ihnen hier ein Malprogramm vor, das derzeit zum Besten und Feinsten zählt, was für den Amiga erhältlich ist: »Deluxe-Paint II«.

Der zunächst auffälligste Unterschied gegenüber der ersten Version ist sicher der, daß nun direkt vom Programm aus die verschiedenen Grafikmodi (Lores, Medres, Hires und Interlace) direkt eingestellt werden können (Bild 1). (Die erste Version verlangte noch die Angabe der Grafikbetriebsart hinter dem Befehl »dPaint«, der vom CLI (= Kommandointerpreter) aus eingegeben werden mußte.)

Außerdem läßt sich jetzt auch direkt angeben, ob das Programm komplett oder nur teilweise geladen wird. Bei letztgenanntem Punkt belegen Programmteile, die nicht laufend gebraucht werden, auch nicht unnötig wertvollen Speicherplatz. Sie werden nur geladen und so lange im Speicher behalten, wie sie auch zur Ausführung benötigt werden.

Hat sich der Anwender nun entschieden, in welchem Modus und mit welchen Programmteilen er arbeiten will, gelangt man nach kurzer Wartezeit, in der noch diverse Dateien geladen werden, in den Zeichenmodus. Dies ist die Hauptarbeitsebene des Programms. Von hier aus lassen sich über eine Menüleiste (am oberen Bildschirmrand) und eine Werkzeugzeile (am rechten Bildschirmrand) alle verfügbaren Funktionen anwählen. Da sich diese beiden Menüleisten sowie der Cursor auf Tastendruck ausblenden lassen (um größere Flächen, die sonst verdeckt wären, zu bearbeiten), sind sinnvollerweise alle nötigen Kommandos jederzeit auch über die Tastatur erreichbar.

Alles, was man braucht . . .

Deluxe-Paint II bietet in vieler Hinsicht Außergewöhnliches. Neben Standardfunktionen wie Linien ziehen, Kreise und Rechtecke zeichnen oder Flächen farbig ausfüllen, lassen sich gepunktete Linien und Geraden in das Bild setzen. Auch Kurvenzeichnen wird unterstützt. Selbstverständlich ist eine Funktion integriert, die im Falle eines Fehlers die letzte Handlung wieder rückgängig macht. Je nach Gegebenheit ist es möglich, verschiedene Pinselformen und -größen auszuwählen. Sogar beliebig große Bildschirmbereiche lassen sich aufnehmen und als Pinsel verwenden. Dazu aber später.

Zuerst einige Bemerkungen zu den normalen, in der Werkzeugzeile erreichbaren Pinseln. Dort finden sich zehn verschiedene Pinselformen, die durch Druck auf die linke Maustaste zum aktuellen Pinsel werden. Drückt man dagegen die rechte Maustaste, kann der ausgewählte Pinsel vor dem Einsatz stufenlos vergrößert oder verkleinert werden. Hier offenbart sich eine weitere Besonderheit, die Deluxe-Paint II von den meisten Programmen unterscheidet: Solange sich der Mauszeiger nicht in der Menüleiste befindet, hat die rechte Maustaste nicht die gewohnte Funktion.

Im Regelfall hat ein Druck auf die rechte Maustaste zur Folge, daß sich die Menüleiste eines jeden Programms (auch der Workbench) einblendet, egal, wo sich der Mauszeiger befindet. Bei Deluxe-Paint II dagegen kann ein Anklicken eines Symbols in der Werkzeugzeile verschiedene Dinge zur Folge haben, abhängig davon, welche Maustaste gerade betätigt wird. Die Menüleiste wird nur aktiviert, wenn sich der Mauszeiger tatsächlich in der Menüleiste befindet.

Um Details eines Bildes zu bearbeiten, kann ein Ausschnitt der Grafik beliebig vergrößert werden (ZOOM). Die Vergrößerung erfolgt (fast) stufenlos. Mit der linken Maustaste wird der ausgewählte Ausschnitt »herangeholt«, mit der rechten Taste »entfernt« (Bild 2). Dies hat den Vorteil, daß, falls man sich nicht mehr sicher ist, an welcher Stelle sich der Ausschnitt im Bild befindet, die nähere Umgebung des Ausschnitts betrachten kann. Auch im Zoom-Modus läßt sich mit den verschiedenen Pinselbreiten arbeiten.

Deluxe-Paint II gestattet es, Text in die Grafik zu setzen. Dabei läßt sich aus der Menüleiste der gerade benötigte Zeichensatz auswählen. Solange noch keine andere Funktion aktiviert wurde, können die einzelnen Buchstaben auch wieder gelöscht werden.

Eine weitere nützliche Hilfsfunktion ist »Grid«. Ist dieses Hilfsmittel aktiv, ist unsichtbar ein freidefinierbares Linienraster über die Grafik gelegt. Der Mauszeiger kann nun nicht mehr frei im Bild bewegt, sondern nur noch auf den gedachten Kreuzungspunkten der Linien positioniert werden. Dadurch läßt sich beispielsweise eine Zeichnung, über die ebenfalls ein Raster gelegt wurde, auf den Bildschirm übertragen. Da Deluxe-Paint II zwei Bildschirme verwaltet, zwischen denen umgeschaltet werden kann, wären so auch Übertragungen zwischen den beiden Grafikseiten denkbar. Doch für diesen Zweck bietet das Programm eine bessere

Lösung. Es läßt sich eine Koordinatenanzeige aktivieren, die jederzeit über die X- und Y-Position des Mauszeigers informiert. Damit wird es ein leichtes, Bildschirmbereiche genau zu bestimmen.

Kommen wir nun zur Eingangs erwähnten Funktion, die es gestattet, beliebig große Bereiche des Bildschirms als Pinsel zu definieren (wir wollen sie zur besseren Unterscheidung im folgenden Bild-Pinsel nennen). Hier bietet das Programm Gigantisches. Denn mit diesen selbstdefinierten Pinseln läßt sich fast alles Vorstellbare machen. Da die Bild-Pinsel beim Umschalten auf die zweite Grafikseite erhalten bleiben, können so Bildteile in das zweite Bild eingesetzt werden. Sehen Sie sich dazu Bild 3 an. Im ersten Grafikschirm hatten wir das Grundgerüst der zu erstellenden Person abgelegt, den Kopf. Oder besser, die Rohform des Kopfes (Bild 3, rechts). In die zweite Grafikseite wurden der Reihe nach einzelne Bilder geladen, die verschiedene »Zusatzteile« des Gesichts (Haare, Augen, Ohren oder Bärte) enthielten. Gerade benötigte Kopfteile konnten als Pinsel definiert und in die erste Grafikseite eingesetzt werden. So entstand mit geringem Aufwand der Kopf eines Mannes (Bild 3, links).

Mit den Bild-Pinseln läßt sich aber noch viel mehr machen. Nachdem erst mal ein Bildteil aufgenommen wurde, läßt es sich um beliebige Winkel drehen, biegen, dehnen, stauchen, verzerren, vergrößern, verkleinern, spiegeln oder auf den Kopf stellen. In Bild 4 links oben sehen Sie das Ausgangsobjekt, ein Flugzeug. Mit sehr wenigen Befehlen und geringem Zeitaufwand entstanden die Abwandlungen dieses Jets.

Die Grafik im Griff

Es gibt noch mehr Möglichkeiten, einen solchen Pinsel zu verändern. Die wichtigste ist aber zweifellos, daß so entstandene Gebilde auf Diskette gespeichert und immer wieder verwendet werden können. Dazu existiert ein eigenes Rollmenü in der Menüleiste, welches sich nur mit den Bild-Pinseln beschäftigt.

Aber auch mit den Farben ist einiges machbar. Deluxe-Paint II erlaubt es, alle vorhandenen Farben zyklisch zu wechseln. Dies bedeutet, daß jede Farbe die der vorangegangenen übernimmt und so weiter. Also kann jede einzelne Farbe auch die Werte aller anderen annehmen. Die Drehung des bekannten Amiga-Balls wurde beispielsweise mit dieser Funktion erzeugt. Das Gebilde in Bild 5 wurde auch mit dieser Methode gemalt. Schaltet man nun die Funktion ein, sieht es aus, als würden sich die Scheiben im Uhrzeigersinn drehen, da jede Farbe auf die Nachbarfarbe übertragen wird.

Doch auch andere Effekte lassen sich mit den Farben erzielen. Weist ein Bild beispielsweise harte Übergänge von einer Farbe zur anderen auf, kann mit einem »Weichmacher« die harte Grenze aufgelockert werden. Ein anderer Effekt ist das Verwaschen von zwei Farben, ähnlich, als würden Sie mit einem Pinsel zwei Ölfarben-Flecke an den Rändern vermischen. Die Funktionen, die sich mit den Farben befassen, sind sehr vielfältig und lassen nichts zu wünschen übrig.

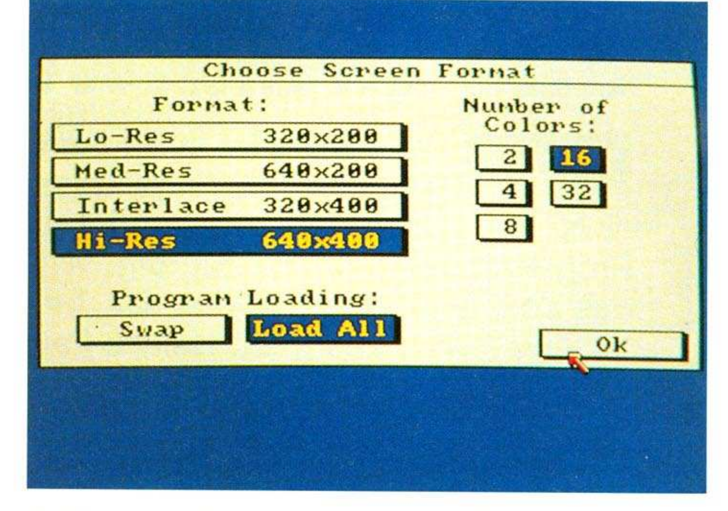


Bild 1. Die auffälligste Neuerung gegenüber Deluxe-Paint I: Die verschiedenen Grafikmodi lassen sich nun direkt vom Programm aus einstellen

Das Programm gestattet die Auswahl von bis zu 32 aus 4096 Farben. Diese Anzahl von Farben dürfte auch für ausgefallene Bilder mehr als ausreichen. Der Amiga bietet zwar die Möglichkeit, im HAM(Hold and Modify)-Modus alle 4096 Farben gleichzeitig darzustellen, jedoch wäre es wenig sinnvoll, diese Palette mit einem Zeichenprogramm zu bearbeiten. Deluxe-Paint II stellt dem Käufer noch weitere praktische Funktionen zur Verfügung. Jeder gezeichnete Punkt kann um beliebig viele Symmetrieachsen gespiegelt werden. Damit sind absolut gleichförmig strukturierte Rosetten oder Spiegelungen kein Problem mehr.

Auch können die Bilder größer sein, als der Bildschirm darzustellen vermag. Für diesen Fall gestattet es das Programm, den gerade sichtbaren Bildausschnitt mit den Cursor-Tasten zu verschieben. Ein spezieller Modus kann auch eine höhere Auflösung anzeigen, damit man sich wenigstens das größere Bild einmal ansehen kann.

Beim Füllen einer Fläche ist man auch nicht allein auf Farben beschränkt. Über eine spezielle Funktion können eigene Füllmuster, die auch aus einem Teil des Bildschirms oder Pinsels bestehen können, entworfen werden. Deluxe-Paint II legt seine Grafiken als IFF(Interchange File Format)-Dateien ab. Es ist damit kompatibel zu vielen weiteren Programmen wie beispielsweise Graphicraft oder Aegis Images. Da man seine Bilder nicht nur gerne auf dem Bildschirm sieht, sondern vielleicht auch zu Papier bringen möchte, ist in Deluxe-Paint II eine sehr kom-

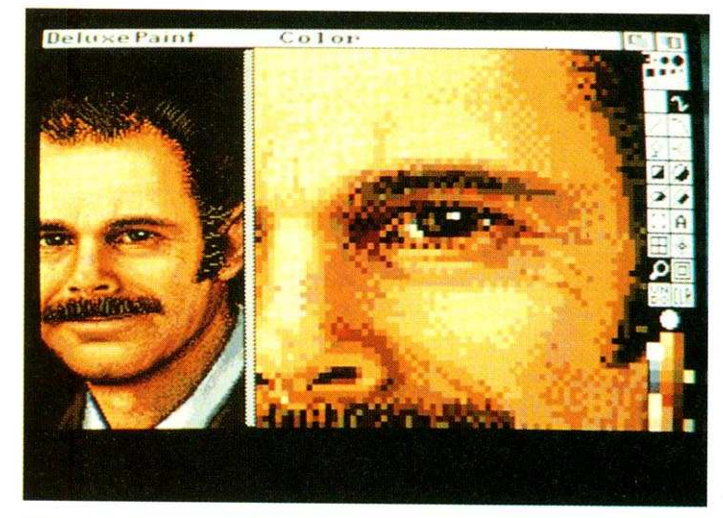


Bild 2. Deluxe-Paint II ermöglicht das stufenlose Vergrößern (»Zoomen«) beliebiger Bildschirmausschnitte

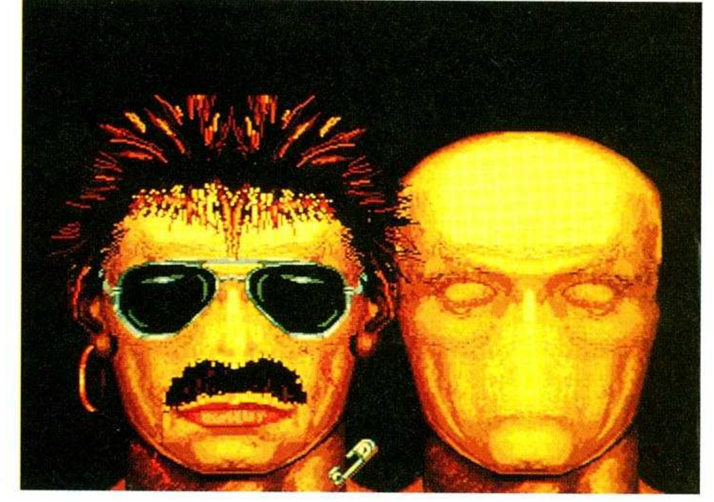


Bild 3. Durch die Verwendung von Pinseln (Brushes) entstand der linke Kopf aus dem rechten Grundmodell

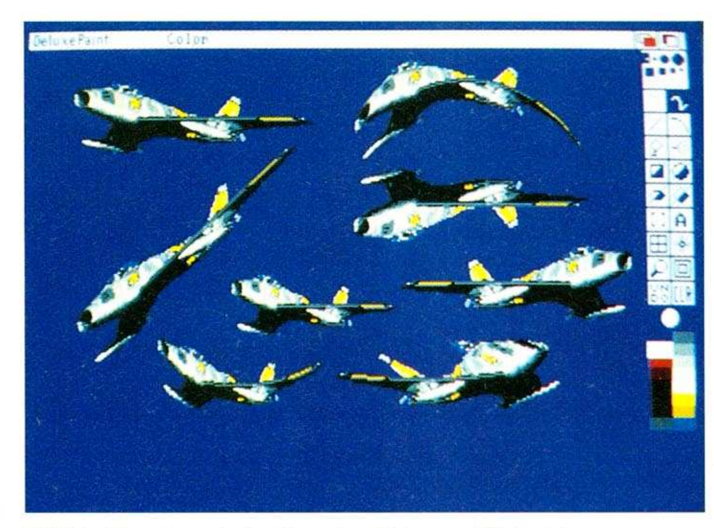


Bild 4. Ein einmal als Pinsel definierter Bildschirmausschnitt kann auf vielfältige Arten verändert werden

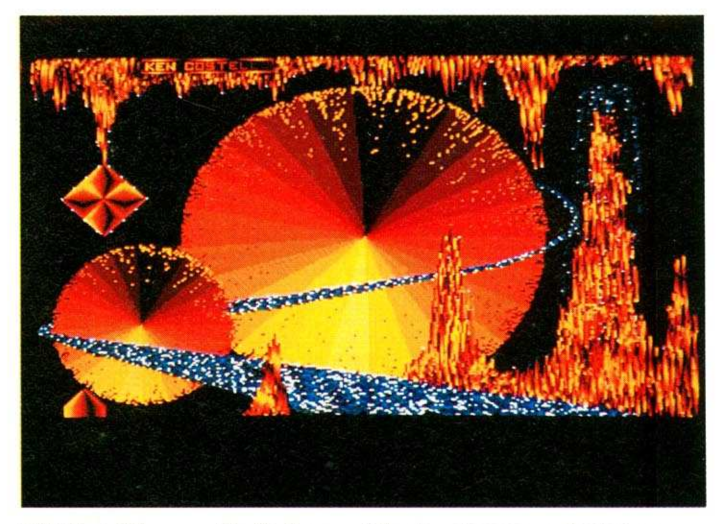


Bild 5. Läßt man die Farben zyklisch rotieren, scheint es, als bewegten sich die Scheiben

fortable Druckerroutine eingebaut. Vor dem Ausdruck erfolgt noch einmal eine Abfrage, in welchem Format und bei welcher Papierlänge das Bild gedruckt werden soll. Ebenfalls läßt sich noch einstellen, ob es eine Schwarzweiß-Hardcopy, ein Ausdruck nach Grauwerten oder die Ausgabe auf einen Farbdrucker sein soll. Auch die Anzahl der Ausdrucke läßt sich bestimmen. Es werden alle Drucker unterstützt, deren Druckertreiber in der Workbench im Programm »Preferences« enthalten sind.

Deluxe-Paint II ist derzeit das leistungsfähigste und am leichtesten zu bedienende Malprogramm, das derzeit für den Amiga angeboten wird. Es bietet alle Möglichkeiten, die dem Anwender das Zeichnen erleichtern. Da es außerdem speziell auf die Hardware des Computers zugeschnitten ist und mit dieser vorbildlich harmoniert, ist es in den meisten Funktionen sehr schnell. Leider gibt es auch einige Punkte, die das ansonsten vorbildhafte Programm trüben:

Es ist leider erforderlich, daß man, wenn man mit der höchsten Auflösung arbeiten will und alle möglichen Farben darstellen möchte, über sehr viel Speicher verfügen muß. 512 KByte sind oft zu wenig. Des weiteren ist das mitgelieferte, sehr gute Handbuch leider nur in englischer Sprache geschrieben.

Wem aber diese Punkte nichts ausmachen, dem sei Deluxe-Paint II auf jeden Fall empfohlen. Der Preis von etwa 280 Mark ist, gemessen an der Leistung und Vielfalt des Programms, nicht zu teuer. Als Alternative dazu bietet sich ja auch noch der Vorgänger Deluxe-Paint I an, der zwar nicht ganz soviel kann, aber mit etwa 220 Mark immer noch zum Besten zählt, was es derzeit an Grafikprogrammen für den Amiga gibt. (dm)

Bezugsquellen: Gut sortierte Software-Anbieter. Die Adressen entnehmen Sie bitte dem Anzeigenteil.

Der Aegis Animator

Grafik und Animation sind Schlagworte, die nicht nur jeden Amiga-Besitzer aufhorchen lassen. Was man alles mit einem guten Animationsprogramm machen kann, erfahren Sie in diesem Bericht.

Zu was eignet sich der Amiga mit seinen vielen Farben und hervorragenden Grafikfähigkeiten besser, als zur Produktion eigener Videosequenzen. Dank der Unterstützung durch die Koprozessoren Agnus und Denise, die speziell für den Amiga entwickelt wurden, ist natürlich die Grafikanimation ein Heimspiel für diesen Computer. Sehr schnell haben die Software-Hersteller auf den Bedarf von Animationsprogrammen reagiert. So zählt Software für dieses Gebiet zu den meistverkauften Utilities überhaupt, denn was ist nützlicher, als die damit ausgearbeiteten Videos für Präsentationen im Heim- oder sogar im kommerziellen Bereich zu verwenden. Schon sehr früh ist aus diesen Überlegungen heraus der Animator von Aegis Development entwickelt worden. Er liegt bereits in einer verbesserten Version vor. Dieser Version wird inzwischen auch noch »Aegis Images«, ein Mal- und Zeichenprogramm, beigelegt, welches früher separat verkauft wurde. Damit erhält man die Möglichkeit, selbst Bilder auf dem Amiga zu malen. Zusatzdisketten mit vorgefertigten Szenen und Objekten sowie einigen nützlichen Tools sind für den Animator extra zu erwerben. Das Programm ist jedoch als Grundsystem schon voll arbeitsfähig, was dadurch bewiesen wird, daß viele Animationen professioneller Spiele schon mit dem Animator produziert wurden. Hauptsächlich wird das System von Sachs Enterprises in den USA benutzt, dessen Chef J.D. Sachs kein Unbekannter mehr in der Computerszene ist. Entwarf er doch schon auf dem Commodore 64 einige Spiele, wie zum Beispiel »Saucer Attack« und andere gute Grafikdemos. Das schwer einzuordnende Spiel »Defender of the Crown«, das Strategie-, Adventure- und Action-Elemente enthält und eine wirklich fantastische, bewegte Grafik auf den Bildschirm des Amiga zaubert, stammt auch aus der Feder dieses Grafikkünstlers. Deshalb soll dieser Animator hier intensiv gestestet werden.

Welche Anforderungen muß man an ein Animationsprogramm für den Amiga stellen? Grundsätzlich gesehen sollte es einfach zu bedienen sein, damit auch Einsteiger sofort Ergebnisse erzielen können und später im normalen Gebrauch nicht zu umständliche Überlegungen nötig sind, um zu einem annehmbaren Resultat zu kommen. Außerdem muß das Programm effizient, schnell und speicherplatzsparend arbeiten. Je klarer die Arbeitsstruktur bei der Produktion der Animationen ist, um so leichter wird es einem fallen, seine Ideen auch wirklich entsprechend umzusetzen; und je schneller das Programm arbeitet, um so mehr Spaß macht es,

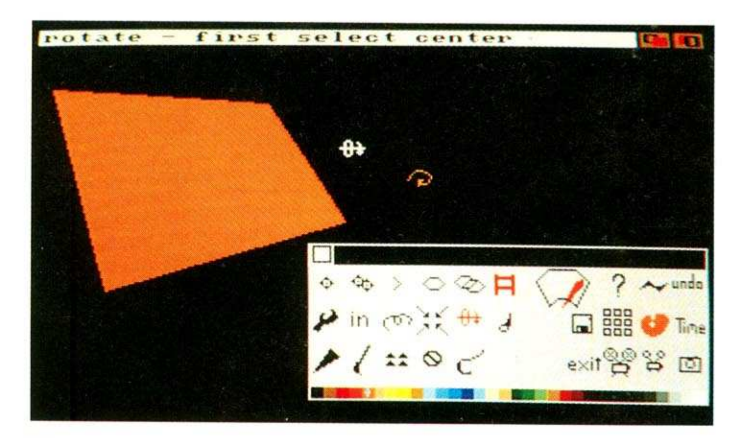


Bild 1. Das »Fast Menu« ermöglicht einen schnellen Zugriff auf die einzelnen Funktionen

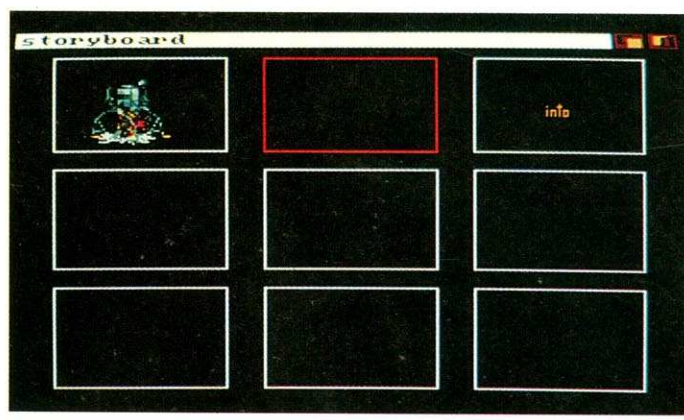


Bild 2. Das Herzstück des Animators ist das Storyboard mit seinen neun »Tweens«

den fertigen Video-Clip anzuschauen. Die Forderung an eine sinnvolle Speicherplatzverwaltung ist sicherlich für alle Programme wichtig, nur kann man sich hier überlegen, daß man um so mehr Animationssequenzen im Computer ablaufen lassen könnte, je mehr Platz dafür vorhanden ist. Da man grundsätzlich nicht davon ausgehen sollte, daß jeder Besitzer eines Amiga diesen mit zusätzlichem Speicherplatz ausrüsten kann und will, muß ein effektiv arbeitendes Programm vorliegen. Zum letzten Punkt ist zu sagen, daß das Hauptprogramm des Animators, welches sich ständig im Speicher des Amiga befindet, ungefähr 150 KByte benötigt. Dennoch bleibt auch auf einem Amiga mit 512 KByte Speicher genügend Platz, um ausreichend lange Sequenzen zu erstellen. Filmreife Videos kann man natürlich nicht vom Animator erwarten, aber das ist auch gar nicht Sinn und Zweck des Programms. Entscheidend ist vielmehr, daß man mit einfachen und auch leicht zu bedienenden Mitteln zu einem brauchbaren Ergebnis kommt. Der Aegis Animator arbeitet mit den üblichen Amiga-Menüs, hat aber zusätzlich noch ein sogenanntes »Fast Menu« (Bild 1). Darin sind alle Funktionen als Icons enthalten, die zum Arbeiten an den einzelnen Sequenzen benötigt werden. Dieses Menü hat einige Vorteile gegenüber den normalen Pull-Down-Menüs. Zum einen ist es ein Fenster, das man beliebig über den Bildschirm verschieben kann und andererseits kann ein Benutzer, sobald er sich mit allen Eigenschaften vertraut gemacht hat, damit schnell und effektiv einzelne Funktionen anwählen. Für Vergeßliche bietet das »Fast Menu« sogar eine Hilfsfunktion an: Beim Anklicken der Icons mit einem Fragezeichen wird ein kurzer Erklärungstext in der Kopfzeile ausgegeben.

Für die einzelnen Szenen eines Videos wird beim Benutzer vorab nur eine ungefähre Vorstellung des Handlungsablaufs bestehen. Es gilt nun, diese Ideen mit Hilfe des Animators umzusetzen. Bei einfachen Szenen, in denen beispielsweise nur ein simples Objekt wie etwa ein Rechteck bewegt werden soll, ist dies sicherlich kein Problem. Der Animator enthält in seinem Create-Menü fünf Punkte um solche Objekte als Drahtgittermodell oder mit ausgefärbten Flächen darzustellen:

☐ Line: ziehen einfacher Linien
☐ Polygon: zum Entwerfen von Drei- oder Vielecken
☐ Circle: für Kreise jeder beliebigen Größe
☐ Star: kreiert sternförmige Objekte
☐ Block: zeichnet Rechtecke.

Wer über diese einfachen Funktionen hinaus eigene Objekte in seinen Videos verwenden möchte, kommt nicht umhin, diese Objekte zu zeichnen. Dies geht nur mit einem leistungsfähigen Zeichenprogramm. Darum hat man dem Animator das Malprogramm Aegis Images beigelegt. Damit können Objekte für Animationen gezeichnet und direkt in den Animator übernommen werden. Da sich die Programme mit ihren Grafiken auch an den IFF-Standard halten, ist es möglich, Grafiken von anderen Zeichenprogrammen wie zum Beispiel dem leistungsfähigeren Deluxe-Paint einzubinden. Images kann man nur bedingt mit Deluxe-Paint vergleichen, aber es ist ja im wesentlichen dazu gedacht, dem Käufer des Animators ein im Preis inbegriffenes Zusatzprogramm in die Hand zu geben, mit dem er selbst kreativ tätig werden kann.

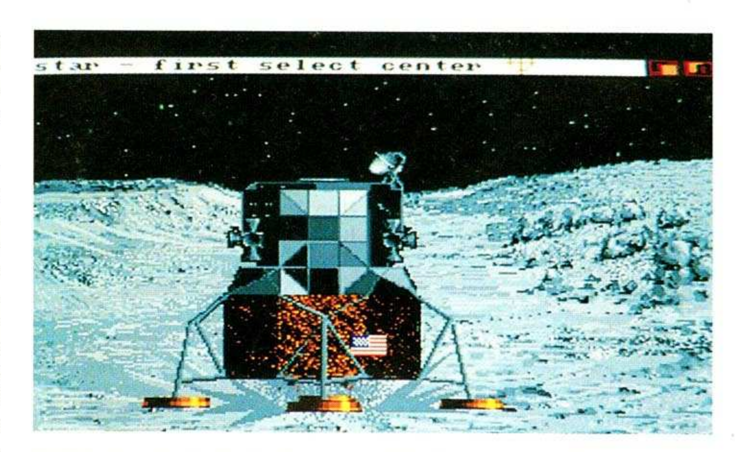


Bild 3. Die komplette Szene mit Hintergrundgrafik aus dem Storyboard von Bild 2

Bewegung bringt Leben

Die Grundwerkzeuge, mit denen die Bewegungen der einzelnen Objekte programmiert werden können, bietet der Animator in seinem Move-Menü. Der einfachste Befehl ist dabei »Move«, mit dem eine horizontale Bewegung vor oder hinter anderen Objekten bestimmt wird. Mit »Rotate« können Objekte um ein beliebig auf dem Bildschirm festlegbares Zentrum gedreht werden. Die Drehachse darf horizontal, vertikal oder senkrecht zur Bildschirmebene liegen. Weiterhin lassen sich mit »Size« Objekte vergrößern oder verkleinern, auch die Farbe und Art (Gitter oder ausgefüllt) von einfachen Objekten darf nachträglich geändert werden. Die stärksten und somit auch am meisten genutzten Befehle sind jedoch »Path« und »Morph«. Bei »Path« läßt sich für die Bewegungsfolge eines Objektes ein Pfad aus Einzelpunkten über den Bildschirm legen, dem dann die Animation getreulich folgt. »Morph« hingegen erlaubt die Dehnung und Verzerrung von Objekten während einer Bewegung. In Zusammenarbeit dieser Kommandos mit einer freien Auswahl, welche Objekte davon betroffen sind, können alle Bewegungen am Bildschirm erzeugt werden. Der eigentliche Animationsvorgang ist dabei denkbar einfach. Nachdem ein geometrisches Modell entworfen oder eine fertige Grafik von Diskette geladen wurde, sucht man sich die gewünschte Bewegungsform aus dem Menü aus und vollzieht die Bewegung mit der Maus direkt am Bildschirm. Man zieht also das Objekt in die gewünschte Richtung oder dreht es um eine Achse. Währenddessen sieht man geometrische Modelle als Drahtgittermodell und nachgeladene Grafiken als ein angedeutetes Fenster. Um vom späteren Aussehen der Szene schon einen Eindruck zu erhalten, werden Hintergrundgrafiken aus Malprogrammen vor-



Bild 4. Jederzeit lassen sich auch im Background die Farben ändern

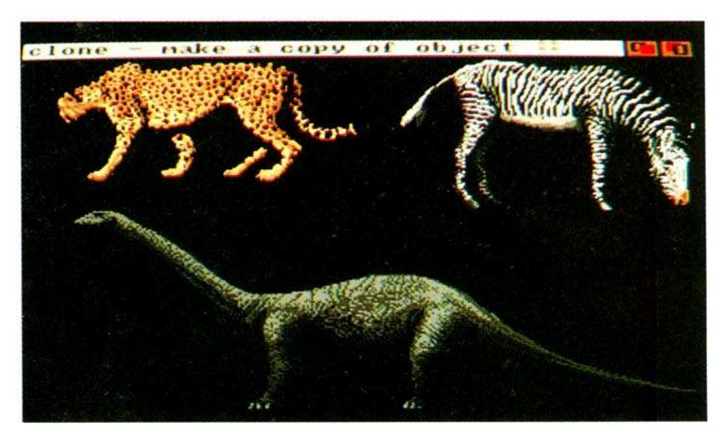


Bild 5. Für alle, die fantastisch gemalte Objekte suchen, gibt es Zusatzgrafiken

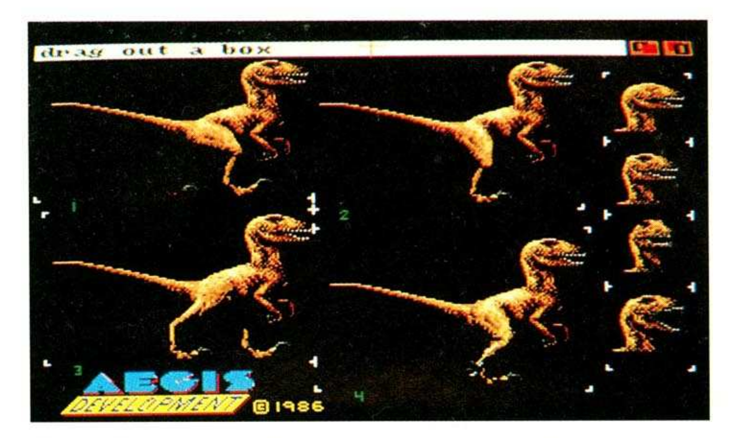


Bild 6. Die einzelnen Bewegungsabläufe des Objektes muß man sich schon vorab überlegen

ab geladen und man kann genau sehen, wo die Bewegungen anzusetzen beziehungsweise zu starten sind. Eine sehr wichtige Eigenschaft bei der Kombination von mehreren verschiedenen Bewegungen ist, daß der Animator bei der Reihenfolge, in der man die Bewegungsphasen eingibt, durchaus unterscheidet. Um dies zu verdeutlichen, hier ein kleines Beispiel: Wir legen für ein Dreieck eine Rotation um eine seiner Spitzen fest und danach einen Pfad aus vielen Punkten wellenförmig über den Bildschirm. Daraus würde eine Bewegung des Dreiecks genau auf dem vorausgezeichneten Pfad erfolgen und gleichzeitig eine Drehung um seine Spitze. Einen ganz anderen Effekt würden wir erzielen, wenn zuerst der Pfad gelegt und danach die Rotation eingegeben worden wäre. Dabei wäre nämlich der Pfad die Grundbewegung an der sich der Animator orientiert, die abhängig von der Lage des Objektes ist. Da eine Drehung um die Spitze des Dreiecks stattfindet, hat, je nach Ablaufgeschwindigkeit der gesamten Szene, das Objekt nach der ersten Phase entsprechend eine andere Lage; es wurde in diesem Fall leicht gedreht. Man muß sich vorstellen, der Verlauf des Pfades ist relativ zum Objekt gesehen aber mitgedreht worden. Schon nach einigen Bewegungsphasen läßt sich ersehen, daß das Objekt den Bildschirm verlassen wird. Dies macht dem Ablauf insgesamt nichts weiter aus, außer daß man eben von dem Dreieck nichts mehr sieht. Um bei Szenen, die viele Objekte verwenden, die Übersicht zu behalten, kann man den gesamten Ablauf in den sogenannten »Ghost Mode« schalten. Darin werden alle Objekte nur als Schattenriß dargestellt, und man kann immer feststellen, wo sie sich gerade befinden. Dies ist vor allem wichtig bei großen Objekten im Vordergrund, die kleinere verdecken.

Kein Film ohne Script

Das Herzstück des Animators ist das Storyboard (Bild 2). Hier werden alle Szenen zusammengeschnitten und in ihrem Gesamtablauf verkettet. Der Animator kann bis zu neun völlig verschiedene Szenen in einen Gesamtablauf bringen. Diese einzelnen Teile werden auch als »Tweens« bezeichnet. Für jede Szene lassen sich natürlich unterschiedliche Hintergrundgrafiken laden und verschiedene Objekte bewegen. Sehr vorteilhaft ist, daß das Storyboard jederzeit eine Abfrage über den momentan noch freien Speicherplatz zuläßt. So kann frühzeitig erkannt werden, wann eine Szene zu groß wird und man besser speichert, um eine neue zu beginnen. Dies geschieht vor allem, wenn zu viele oder zu große Objekte verwendet wurden. Witzig ist nebenbei bemerkt die Meldung, die der Animator ausgibt, falls man ohne es mitzubekommen, die Speichergrenze überschritten hat: »Memory Panic, System going down«. Glücklicherweise dürfen alle erstellten Änderungen noch schnell auf Diskette gespeichert werden, bevor man die Szene verliert. Gut ist auch, daß in den kleinen Fenstern des Storyboards die Szenen mit Animation einzeln oder zusammen gestartet und betrachtet werden können. Man sieht zumindest die Objekte und ihre Bewegungen, der Hintergrund ist nur im Großbild zu sehen (Bild 3). Einziger Schönheitsfehler des Animators ist, daß sich keine Musik oder »gesampelte« Geräusche in Videos einbinden lassen; es bleibt also bei einem stummen Spaß. Sehr vorteilhaft ist allerdings, daß schon Zusatzdisketten mit Hintergrundgrafiken (Bild 4), Objekten (Bild 5) und vorbereiteten animationsfertigen Sequenzen (Bild 6) vorliegen. Das einzige, was einen noch davon abhalten könnte, sich dieses leistungsfähige Programm zuzulegen, ist vielleicht der recht hohe Preis von 398 Mark für das Grundpaket mit Animator und Images. (jk)

Bezugsquelle: Alle gutsortierten Softwareanbieter. Einzelne Adressen entnehmen Sie bitte dem Anzeigenteil.

True Basic — ein Hauch von Pascal

Basic ist bei Einsteigern und Fortgeschrittenen die wohl bekannteste Programmiersprache. Von vielen professionellen Programmierern wegen seiner Unstrukturiertheit verpönt, ist Basic in einem Wandel begriffen, einem Wandel zu einer perfekten Programmiersprache.

Das Wort »Basic« ist die Abkürzung für »Beginners All Symbolic Instruction Code« und zeigt, wenn übersetzt, den ursprünglichen Zweck dieser Programmiersprache. Sie soll Anfängern die Arbeitsweise eines Computers auf einfache Weise näherbringen. So wird fast jeder Computer-Einsteiger zunächst Basic erlernen, um mit seinem Computer zu kommunizieren. Wer jedoch längere Zeit mit dem Basic des C 64 arbeiten mußte, wird sich schnell der Unzulänglichkeiten dieser Sprache bewußt. Wenige oder gar keine Regeln lassen Basic-Programme bald zu einer wirren Ansammlung von Befehlen werden, deren logische Zusammenhänge kaum mehr entziffert werden können. Insbesondere bietet das Basic 2.0 des C 64 einen nur sehr geringen Befehlsumfang und läßt auch in der Abarbeitungsgeschwindigkeit (Interpreter) zu wünschen übrig.

Einen wesentlichen Fortschritt tat da der C 128 mit Basic 7.0 und der kleineren Version Basic 3.5, die in die kleinen Computer C 16 und Plus/4 implementiert wurde. Eine Vielzahl neuer Befehle erleichtern die Handhabung von Grafik und Sound. Auch die Möglichkeit der Strukturierung von Programmen wurde mit Strukturanweisungen wie DO.. LOOP, WHILE oder UNTIL erheblich verbessert.

Basic auf dem Amiga

Angesichts dieser Entwicklung ist es nicht verwunderlich, daß sich nach Erscheinen des Personal Computers Amiga von Commodore die Basic-Entwickler (John G. Kemeny und Thomas E. Kurz) daran machten, neben dem Amiga-Basic eine für diesen Computer gerechte Version der so beliebten Sprache zu entwickeln. Das Resultat lautet »True Basic«, das ursprünglich für andere Personal Computer geschrieben wurde. Seit kurzer Zeit ist die Amiga-Version auch in Deutschland erhältlich. Als »Language System for the Future« (Sprachsystem für die Zukunft) soll sie richtungsweisend für die Zukunft von Basic sein. Wir zeigen Ihnen, wie diese Zukunft aussieht.

Hat man ein Exemplar von »True Basic« erstanden, fällt zunächst das Gewicht und die Größe der Verpackung auf. Geschützt durch einen kräftigen Pappbehälter findet man neben der Programmdiskette zwei Handbücher, die beide etwa 300 Seiten umfassen. Doch sollte man angesichts der umfangreichen Literatur nicht erschrecken. Das erste Buch mit der Aufschrift »Reference Manual« befaßt sich mit den allgemeinen Eigenschaften und Fähigkeiten von True Basic. Es liefert einen schnellen Überblick über den Befehlsumfang und seine Besonderheiten. Es soll als Nachschlagewerk für spätere Programmierarbeiten dienen.

Das zweite Buch, der »User Guide,« wendet sich an die Einsteiger, die intensiv Basic erlernen möchten. Ähnlich einem Lehrbuch wird der Neuling anfangs mit den Grundsätzen der Basic-Programmierung vertraut gemacht. Die anschließenden Kapitel bauen auf die zuvor erworbenen Kenntnisse auf und ermöglichen so Schritt für Schritt die Entwicklung vom Anfänger bis hin zum perfekten Basic-Programmierer. Während sich das »Reference Manual« nur auf True Basic allgemein bezieht, befaßt sich der »User Guide« direkt mit der Arbeit auf dem Amiga. So wurde ein umfassendes Kapitel für die Bedienung von True Basic auf dem Amiga reserviert.

Dem Studieren dieser Bücher steht allerdings ein Hindernis im Weg. Sämtliche Begleitliteratur von True Basic ist in Englisch verfaßt, und mag aus diesem Grund manchen davon abhalten, die beiden Handbücher ausgiebig zu lesen.

Das Englisch beider Handbücher ist jedoch vorzüglich und relativ einfach gehalten, so daß auch geringere Englischkenntnisse den Lesefluß nicht behindern. Zudem sorgen große Buchstaben und außergewöhnlich viele Beispiele und Grafiken für genügend Auflockerung.

Bild 1. Die Benutzeroberfläche von True Basic — auf Wunsch alles auf einen Blick

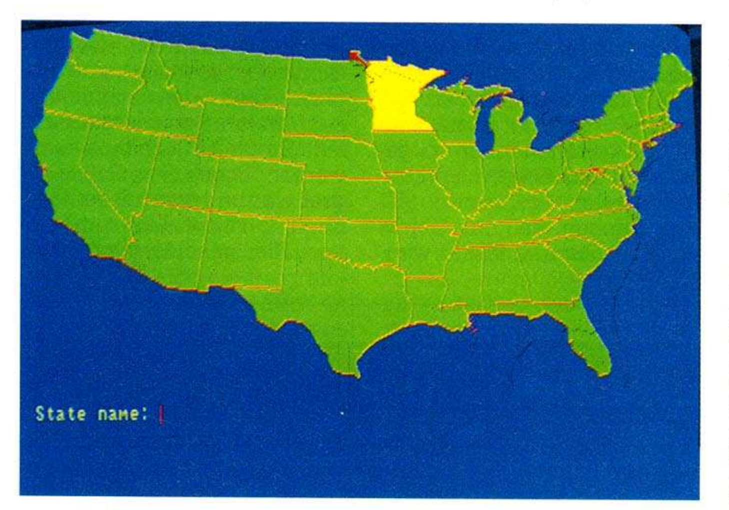


Bild 2. Schon die Demoprogramme lassen die Grafikfähigkeiten von True Basic erahnen

Amiga-typische Benutzeroberfläche

Nach dem Laden von True Basic fällt sofort auf, daß alle Aktionen über die vom Amiga bekannten Pull-Down-Menüs gesteuert werden. Klickt man mit der Maus die obere Bildschirmleiste an, hat man eine Auswahl verschiedener Punkte, die Kommandos zum Editieren und Bearbeiten von Basic-Programmen enthalten. Der verbleibende Bildschirmbereich steht für die Eingabe von



Markt&Technik
68000er
DAS MAGAZIN DER NEUEN COMPUTER-GENERATION
FEBRUAR 1987
2/87
DM 7,-
Atari ST
Amiga
Grafik-Sensation
Deluxe Paint II
QL
Große Assembler-Parade
Mac
kreativ!
Interview mit »Axel F.« – Komponist Harold Faltermeyer
Viele anspruchsvolle Listings für Atari ST, Amiga und QL
AUFBRUCH IN EINE NEUE DIMENSION
mit »68000er«, dem Magazin der neuen Computer-Generation
▶ Programmiersprachekurse für Basic, C, Modula und Assembler.
▶ Bauanleitungen für professionelle Hardware-Erweiterungen.
▶ Spiele-Spaß und -Spannung auf höchstem Niveau.
Ihre hot-line zur Spitzentechnologie von Atari ST, Amiga, Macintosh und Sinclair QL.
Das »68000er«-Magazin erscheint
jeden Monat neu!
POSTER & GUTSCHEIN
KOSTENLOS FÜR SIE
84 mal 60 Zentimeter High-Tech-Szene erwarten Sie! Ihr »68000er«-Poster ist im Abonnementpreis enthalten und gehört Ihnen, auch wenn Sie Ihre Bestellung widerrufen sollten.
FÜR EIN KOSTENLOSES PROBEEXEMPLAR DES »68000er«-MAGAZINS
JA, ich möchte »68000er«, das Magazin der neuen Computer-Generation, kennenlernen.
Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »68000er« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte es dann regelmäßig frei Haus per Post. Außerdem nutze ich den Abonnement-Preisvorteil von 8% und bezahle pro Jahr nur 77,– DM statt 84,– DM im Einzelverkauf.
Vorname
Name
Straße
PLZ, Ort
Datum
1. Unterschrift
Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.
Datum
2. Unterschrift
Gutschein ausfüllen und absenden an: Markt & Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar
A6/7/87

AMIGA TEST

Programmen zur Verfügung. Die meisten Kommandos von True Basic lassen sich einfach mit der Maus erreichen. Auf Wunsch können sie jedoch auch von Hand, sozusagen im Direktmodus, eingegeben werden. Dazu öffnet man ein weiteres Window mit dem Namen »Command« (Bild 1), das die direkte Eingabe erlaubt. Dort werden auch sämtliche Systemmeldungen von True Basic, wie zum Beispiel Fehlermeldungen angezeigt. Hat man das Kommando-Fenster nicht aktiviert, werden die Systemmeldungen in der untersten Zeile des Bildschirms ausgegeben.

Mit dem Handbuch neben dem Computer kann man nun die ersten Basic-Programme schreiben und mit Hilfe der Menü-Kommandos editieren. Die Möglichkeiten reichen hier vom einfachen Löschen einzelner Zeilen oder ganzen Blöcken, dem Verschieben oder Duplizieren von Programmteilen bis hin zum Suchen und automatischen Ersetzen von Zeichenketten. Daneben kann mit einem einfachen Mausklick ein externes Basic-Programm von Diskette in den aktuellen Text eingebunden werden. Ein spezieller Edit-Befehl erlaubt das unabhängige Editieren von eigens augewählten Programmsequenzen. Diese Art der Teil-Editierung ist besonders bei großen Basic-Programmen von Vorteil, wenn man beispielsweise nur ein bestimmtes Unterprogramm verändern will. Die wichtigsten Kommandos lassen sich auch über die Amiga-Taste erreichen, so daß man auch gänzlich ohne Maus arbeiten kann. Hat man zusätzlich das »Command«-Window geöffnet, dürfen die Befehle auch direkt eingegeben werden.

Weitere Kommandos, die nicht über die Menüs zu aktivieren sind, sind über die Funktionstasten zu erreichen. Damit können schnell und problemlos Anfang und Ende des Programms gefunden oder einzelne Worte und Zeilen gelöscht werden. Eine »UNDO«-Funktion erlaubt sogar das Rückgängigmachen von Fehleingaben.

Trotz der umfangreichen Editier-Befehle lassen sich schon nach kurzer Zeit Basic-Programme schreiben und austesten, wobei besonders der klare Aufbau der Pull-Down-Menüs ausschlaggebend ist.

Hat man ein Programm mit dem Editor geschrieben, kann es mit RUN gestartet werden. Man wird bemerken, daß je nach Länge des Programms eine gewisse Verzögerung eintritt, bis das Programm startet. Dies liegt an einer herausragenden Besonderheit von True Basic. True Basic ist kein Interpreter, sondern ein Basic-Compiler, der stets bei Eingabe von RUN aktiviert wird. Der Geschwindigkeitsvorteil gegenüber Interpreter-Systemen ist enorm. True-Basic-Programme sind deshalb sehr schnell, was einige kleine Testprogramme eindrucksvoll beweisen. Tritt während der Compilation ein Fehler auf, weist eine der vielen Fehlermeldungen auf die Ungereimtheit hin, während der Cursor auf die unkorrekte Stelle plaziert wird. Nach der Korrektur des Fehlers startet man erneut mit RUN. Die Fehlersuche ist somit recht bequem.

Nach der Fertigstellung eines Programms kann es schließlich mit »COMPILE« endgültig compiliert und in dieser Form auf Diskette gespeichert werden. Dieses Compilat ist zwar nicht außerhalb der True-Basic-Oberfläche ablauffähig, wird aber beim Start nicht mehr übersetzt, sondern sofort gestartet. Um ablauffähige Compilate zu erzeugen, wird eine Erweiterung angeboten, die zum Compilat die nötigen Ablaufmodule hinzufügt.

Doch neben dem vorbildlichen Editor ist bei einer Programmiersprache selbstverständlich der Sprachumfang das wichtigste Kriterium.

Ist das noch Basic?

Der Sprachumfang ist bei True Basic ungewöhnlich groß. Ebenso wie dessen Eigenheiten. Schon beim Lesen der ersten Zeilen der Handbücher wird man vor die Tatsache gestellt, daß True Basic gänzlich von Zeilennummern abgekommen ist. Lediglich aus Gründen der Kompatibilität zu anderen Basic-Dialekten ist die Arbeit mit Zeilennummern noch erlaubt. Selbst die, laut Handbuch veralteten, Anweisungen GOTO und GOSUB in Verbindung mit Zeilennummern werden von True Basic akzeptiert. Der zukünftige True-Basic-Programmierer wird jedoch bald auf diese verzichten, wenn er erfährt, welche programmtechnischen Möglichkeiten ihm zur Verfügung stehen.

Variablen und Rechengenauigkeit

Konstante, Variable und Unterprogramme erhalten in True Basic einen Namen, der jeweils bis zu 31 Zeichen lang sein darf. Variablen mit abstrakten Namen wie zum Beispiel »EZ« oder »A« gehören damit der Vergangenheit an. Numerische Variablen arbeiten mit einer Rechengenauigkeit von 10 Stellen und dürfen sich in einem Bereich von 1e-99 bis 1e+99 bewegen. Strings können eine Länge von bis zu 32000 (!) Zeichen annehmen.

Gerade im Bereich der numerischen Datenverarbeitung zeigt sich True Basic als wahrer Rechenkünstler. Eine umfangreiche Sammlung an fest eingebauten Funktionen machen das Berechnen eines Logarithmus (Basis e,2 und 10) oder Sinus zum Vergnügen. Haben Werte zu viele Nachkommastellen, können sie problemlos gerundet oder abgeschnitten werden. Das Besondere bei den trigonometrischen Funktionen ist die beliebige Wahl zwischen Neugrad oder Bogenmaß.

Sollte die Sammlung an mathematischen Funktionen nicht ausreichen, kann man schnell und einfach auf eine Funktionsbibliothek (Library) auf der Systemdiskette zurückgreifen, die weitere Rechenfunktionen bereitstellt. Sie enthält Funktionen wie zum Beispiel Cotangens oder Sekans. Selbst Funktionen wie die Normal-, Binominal- und Poissonverteilung fehlen nicht.

Strings — kein Problem

Bei der Bearbeitung von Strings geht True Basic weiter als manch andere Basic-Systeme. So stehen neben den bekannten Befehlen wie LEN, CHR$ oder STR$ Anweisungen zum Durchsuchen oder Formatieren eines Strings zur Verfügung. Weitere Befehle entfernen überflüssige Leerzeichen, wie sie etwa bei Texteingaben entstehen können. Die Statements LEFT$, RIGHT$ und MID$, wie man sie vom C 64 kennt, sind in True Basic durch ein einfacheres Anweisungsprinzip zusammengefaßt. Parameterangaben hinter dem Variablennamen ermöglichen das Herausgreifen einzelner Teile eines Strings. Die ersten fünf Zeichen einer Variable mit dem Namen »TEST« können beispielsweise so selektiert werden:

```
test$="beispiel"
substring$=test$(1:5)
```

Ebenso können Zeichenketten auf einfache Weise in einen String eingefügt werden:

```
test$(1:0)="ein "
```

ergibt für »test$«

```
"ein beispiel"
```

Die Möglichkeiten scheinen unbegrenzt. Für die Ein- und Ausgabe von numerischen oder Stringvariablen sind in True Basic verschiedene INPUT und PRINT-Anweisungen vorgesehen. So gibt es beispielsweise den Befehl INPUT LINE zur komfortablen Eingabe oder die bekannte Anweisung PRINT USING zur formatierten Ausgabe.

Auch die Standardbefehle READ, DATA und RESTORE fehlen nicht. Hier zeigt sich jedoch ein Nachteil. Leider kann mit RESTORE nur an den Anfang der gesamten DATA-Liste gesprungen werden. Eine eventuelle Angabe einer Zeilennummer oder eines Labels ist nicht vorgesehen.

Matrizen-Jongleur

Der wohl mächtigste Befehl ist die Anweisung MAT, die in Kombination mit vielen Befehlen verwendet werden darf. Sie bewirkt ein zusammengefaßtes Abarbeiten von Feldern. Soll zum Beispiel ein Datenfeld auf dem Bildschirm ausgegeben werden, ist in True Basic eine FOR-NEXT-Schleife unnötig. Man tippt lediglich:

```
mat print feld,
```

Das gesamte Feld »feld« wird somit tabellarisch auf dem Monitor sichtbar. Selbst eine Begrenzung auf bestimmte Elemente ist möglich:

```
mat print feld(5 to 20),
```

Will man DATA-Werte in ein Feld einlesen, genügt:

```
mat read feld
```

Daneben können zwei Felder addiert oder subtrahiert oder eine Skalarmultiplikation durchgeführt werden.

Die hier erwähnten Funktionen des MAT-Befehls sind nur

ein kleiner Teil der denkbaren Anwendungen. Besonders in Zusammenhang mit einigen Grafikbefehlen entfaltet er seine volle Flexibilität.

Tolle Grafik, aber dünne Musik

Kommen wir nun zu den grafischen und musischen Fähigkeiten von True Basic, die angesichts des bisher enormen Befehlsvorrates einige Erwartungen wecken.

Dem Grafikprofi Amiga angemessen, bietet True Basic eine Vielfalt an Grafikbefehlen. Anweisungen zum Setzen von Punkten und Zeichnen von Linien sind dabei obligatorisch. Aber auch Rechtecke, Kreise oder Ellipsen, die auf Wunsch mit der aktuellen Farbe ausgefüllt werden, sind für True Basic kein Problem (schnelle Ausführung). Bemerkenswert ist die Anweisungsfolge SET WINDOW, die es erlaubt, den Bildschirm in X- und Y-Richtung beliebig zu skalieren. Der Definitions- und Wertebereich des Koordinatensystems ist dadurch frei wählbar. Komplizierte Umrechnungen von Programmdaten auf die Punktkoordinaten sind deshalb nicht nötig.

Eine Besonderheit von True Basic ist die Möglichkeit, mehrere Grafikanweisungen zu einem Bild zusammenzufassen, das schließlich in seiner Gesamtheit mit einem Namen aufgerufen werden kann. Zusätzliche Funktionen, die mit dem Bindewort »with« an die Zeichenanweisung angehängt werden, verändern das Bild (Picture) nachträglich. Bilder lassen sich verschieben, verzerren oder können um den Ursprung des Koordinatensystems rotieren. Mit der BOX-Anweisung gezeichnete Körper, wie Rechtecke oder Ellipsen, werden allerdings von der Rotations-Funktion ignoriert.

True Basic unterstützt die unterschiedlichen Grafikmodi des Amiga. Mit der Option SET MODE lassen sich sämtliche Auflösungen von 320 x 200 Punkten mit zwei Farben bis hin zum Interlaced-Modus mit einer Auflösung von 640 x 400 Punkten bei 16 Farben aktivieren. Die Bedienung der Grafik bleibt in jedem Modus gleich bequem (Bild 2). Um die maximal 16 Farben anzusteuern, besitzt True Basic einen Farbbefehl, der neben den Farbnummern auch die Angabe der Farbnamen erlaubt. Hinzu kommt eine Anweisung, deren Anwendung die Intensität der drei Farbbestandteile (Rot, Blau, Gelb) verändert, so daß bis zu 4096 Farbkombinationen erreichbar sind.

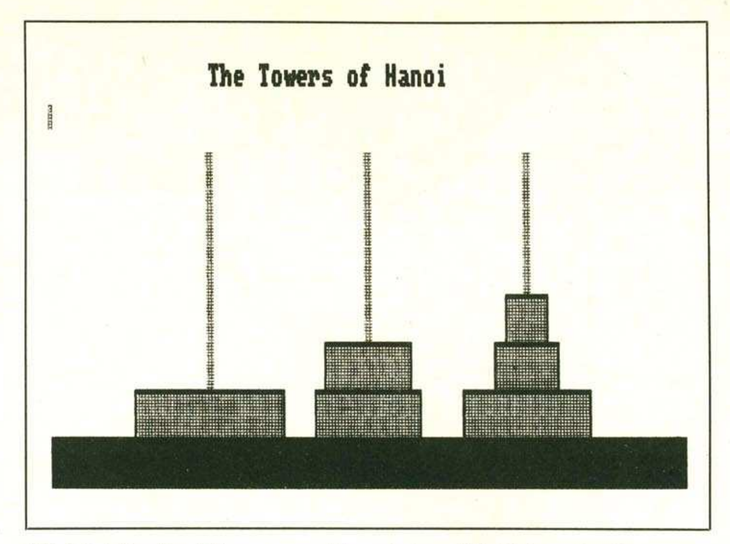


Bild 3. Für die »Türme von Hanoi« benötigt True Basic nur 3,19 Sekunden

Betrachtet man nun die Anweisung zur Musikerzeugung, sollte man seine Erwartungen drastisch verringern. Als man True Basic, das ursprünglich für Personal Computer konzipiert war, für den Amiga umsetzte, hat man offensichtlich versäumt, Anpassungen an die außergewöhnlichen Musikfähigkeiten des Amiga zu implementieren. Effektiv stehen nur zwei Befehle zur Erzeugung von Tönen zur Verfügung. Mit PLAY können in Form von Strings einfache Melodien in Notenschreibweise erzeugt werden. Obwohl viele Optionen, wie zum Beispiel das interruptgesteuerte Abspielen von Musik, integriert sind, werden die Soundmöglichkeiten des Amiga nicht im geringsten ausgenutzt. Melodien können nur einstimmig programmiert werden, wobei diese Stimme nicht einmal veränderbar ist. Die Nutzung der weiteren drei Stimmen und die Klangsynthese, die beim Amiga möglich ist, sind True Basic gänzlich unbekannt. Lediglich der Befehl SOUND gestattet es, auf mehr oder weniger umständliche Weise etwa Geräuscheffekte zu programmieren. Er spielt einen Ton in der angegebenen Frequenz und Zeitdauer. Die Produktion von komplexen Sounds ist damit aber nicht denkbar.

Zur Strukturierung und logischen Regelung all der erwähnten Befehle bietet True Basic Steuer-Anweisungen, die an die Programmiersprache Pascal erinnern.

Neben der IF-THEN-ELSE-Konstruktion geben Anweisungen wie DO. . LOOP, WHILE oder UNTIL dem Programmierer Werkzeuge in die Hand, seine Programme sinnvoll zu strukturieren.

Struktur im Programm

Unterprogramme, die in Subroutinen und Funktionen aufgeteilt sind, erhalten in True Basic Namen, anhand derer sie jederzeit aufgerufen werden können. Diese können dabei intern, das heißt im Programm stehen, oder als externe Unterroutinen gelten, die aus Funktionsbibliotheken auf Diskette stammen. Auf einigen Umwegen lassen sich sogar C-Programme oder entsprechende Maschinenprogramme als externe Basic-Funktionen deklarieren, um im Programm verwendet zu werden.

Für Basic ungewöhnlich ist dabei die Unterscheidung von globalen und lokalen Variablen. Innerhalb eines Unterprogramms definierte Variable sind nur diesem zugänglich und können von anderen Programmteilen nicht berührt werden. Globale, das heißt im Hauptprogramm deklarierte Variablen sind dagegen jeder Unterroutine bekannt.

True Basic unterstützt damit die strukturierte Programmierung und die Modularisierung von Programmen in dem Maße, wie sie zuvor nur von Pascal oder C bekannt war.

Selbstverständlich stellt True Basic auch für die Datenspeicherung auf Diskette einen größeren Umfang von Befehlen zur Verfügung. Der Programmierer hat hier die Wahl zwischen Text-, Record- und Bytefiles, wobei Record-Files die komplexeste Struktur besitzen. Hier können unterschiedliche Datenformen schnell auf Diskette abgelegt und ebenso schnell wieder gefunden werden.

Licht und Schatten

Die Fähigkeiten von True Basic sind damit noch nicht erschöpft. Wir haben hier lediglich die wichtigsten Aspekte dieses Basic-Systems beleuchtet.

Trotz der Befehlsvielfalt, die für Basic recht ungewöhnlich ist, nutzt True Basic die Eigenschaften des Amiga nicht vollends aus. Wir haben die ungenügenden Möglichkeiten der Sounderzeugung bereits angesprochen. Schön wäre auch der Einsatz von interruptgesteuerten Unterprogrammen, wie sie auf Basic-Systemen anderer Computer bereits verwirklicht sind. Dieser oder jener Mangel darf aber nicht darüber hinwegtäuschen, daß es sich bei True Basic um ein überaus leistungsfähiges Basic-System handelt, das besonders durch seinen Bedienungskomfort und der Ausführungsgeschwindigkeit der Programme bestícht (Bild 3). True Basic zeigt, daß die Programmiersprache Basic durchaus professionellen Charakter annehmen kann. Wer bisher mit dem etwas mageren Basic des C 64 gearbeitet hat, wird auf jeden Fall von den Fähigkeiten von True Basic überzeugt sein. Gegenüber dem mitgelieferten Amiga-Basic bietet True Basic den Vorteil einer schnelleren Ablaufgeschwindigkeit der Programme und einen größeren Sprachumfang. Sollten Sie sich mit dem Gedanken tragen, eines Tages auf die Sprache Pascal umzusteigen, haben Sie mit True Basic ein Produkt in der Hand, das die wichtigsten Merkmale von Pascal bereits implementiert hat. In der Tat also ein Basic-Interpreter, der, wie die Bezeichnung bereits verheißt, den Weg aufzeigt, den Basic in Zukunft gehen wird. Weg von unleserlichem Spaghetti-Code, dafür hin zur allseits gepriesenen strukturierten Programmierung, die damit auch den bekanntesten aller Programmiersprachen erschlossen wird.

(Michael Thomas/rf)

Info: Philgerma GmbH, Ungererstr. 42, 8000 München 40, Preis: 349 Mark
Softwareland, Franklinstr. 27, CH-8050 Zürich

Tips zu Programmiersprachen . . .

. . .werden Sie an dieser Stelle in jeder Ausgabe finden. Kleine Utility-Programme, die Ihnen, nicht zuletzt durch den dokumentierten Quellcode, sicher nützlich und lehrreich sind.

Diesmal haben wir für Sie einige kleine C-Programme, die Sie bei der Erstellung eigener Listings und der Arbeit mit dem CLI unterstützen, ausgewählt.

FTYPE — Drucken mit Komfort

Bei FTYPE (Listing 1) handelt es sich um ein kleines C-Programm, um Dateien auf Drucker auszugeben. Nichts Besonderes, meint vielleicht jetzt der eine oder andere, das funktioniert doch auch mit einer Ausgabeumlenkung mit Hilfe des CLI. Die Leistungsfähigkeit dieser CLI-Erweiterung wird Sie jedoch sehr schnell überzeugen. FTYPE druckt nicht nur einfach die Daten, wie Sie zum Drucker gesendet werden, es lassen sich jede Menge Parameter mit übergeben. Das Ganze geht so weit, daß man am Drucker selbst keinerlei Probleme mehr mit DIP-Schaltern oder anderen Bedienungselementen hat. Alles wird von FTYPE, je nach Einstellung, automatisch übernommen. Neben einer indirekten Programmierung des Druckers lassen sich noch weitere Funktionen anwenden. So können Sie Ihre Dokumente wahlweise mit Seiten- und Zeilennummern versehen oder den Seitenvorschub regeln. Doch nun zur genauen Bedienung des Programms:

```
Ftype Datei Zeilenzahl -pnl cmd_1 cmd_2 cmd_3 cmd_4
cmd_5 cmd_6
```

Alle übergebenen Parameter können wahlweise angegeben werden, sind also nicht zwingend erforderlich. Die Zeilenzahl bestimmt die Anzahl der Zeilen pro Seite. Vor den nun folgenden Parametern »pnl« muß ein Bindestrich stehen. »p« schaltet dabei die automatische Seitennumerierung an. Der Parameter »n« aktiviert die Zeilennumerierung. Wird »l« mitangegeben, führt der Drucker am Seitenende keinen Formfeed, sondern einen Linefeed aus. Danach lassen sich bis zu sechs Kommandos in Form von Zahlen an den Drucker senden. Die Zahlen entsprechen dem jeweiligen Befehlscode des ESC/P-Standards. Wollen Sie also beispielsweise die Datei »Test.Txt« drucken, mit einer Zeilenanzahl von 66, sowie mit deutschem Zeichensatz, 137 Zeichen pro Zeile mit vorhergehendem Drucker-Reset, so benötigen Sie dafür folgenden Aufruf für FTYPE:

```
ftype test.txt 66 -pn 0 36 17
```

Dabei werden dann noch die Zeilen wie auch die Seiten automatisch durchnumeriert. Tabelle 1 gibt Auskunft über die zur Compilation notwendige Konfiguration. (Michael Rippl/rf)

Programmname:	FTYPE V 2.2
Sprache:	C
Compiler:	Lattice C V 3.03

Tabelle 1. Steckbrief zu FType

INITCLI für Individualisten

INITCLI (Listing 2) eignet sich besonders für Batch-Dateien, da durch den Aufruf eines einzigen Programms bereits für den Anwender wichtige Parameter eingestellt werden. So läßt sich festlegen, ob Sie im Monochrom-Modus (Grünbildschirm) oder mit einheitlichen Farben, beispielsweise weiße Schrift auf blauem Hintergrund, arbeiten wollen. Dies erfolgt über das Aktivieren oder Deaktivieren der Solobitmap. Ein Aktivieren spart zusätzlich 8 KByte, bei gleichzeitig aktiviertem Interlaced-Modus sogar 16 KByte.

Hier die einzelnen Parameter im Überblick:
? = Aufruf einer Bedienungsanleitung
-s = schaltet Solobitmap aus
-S = schaltet Solobitmap ein
-c = schaltet die durch Preferences vorgegebenen Farben ein
-C = schaltet die Monochrom-Darstellung ein
-w = vergrößert das CLI-Fenster auf die maximale Ausdehnung
-i = schaltet den Interlaced-Modus aus
-I = schaltet den Interlaced-Modus ein

Das Programm wurde mit dem Aztec C-Compiler (Tabelle 2) erstellt und kann ohne die Option »+L« compiliert werden. Benötigt wird lediglich die Bibliothek »c.lib« beim Linken.

(Torsten Juergleit/rf)

Programmname:	INITCLI
Sprache:	C
Compiler:	Aztec C

Tabelle 2. Konfiguration von INITCLI

Menüs in C

Der Aufsteiger von Basic zu C muß mit Schrecken feststellen, daß es keinen Menü-Befehl mehr gibt. Diese Lücke schließen die vorliegenden Routinen (Listing 3), die für den Lattice C-Compiler geschrieben sind. Die so zur Verfügung stehenden Menüs können Sie in eigenen Programmen ohne Probleme einsetzen. Alle Menü-Routinen benötigen als Unterprogramm eine Routine zur Bestimmung der Länge der Strings (Listing 4). Listing 5 zeigt Ihnen ein kleines Demo, in dem die Menü-Routinen angewendet werden.

Doch nun zum Programm selbst. Alle Menüs können auch über die Tastatur bedient werden. Man bezeichnet dieses Verfahren als Shortcut (Rechte AMIGA-Taste plus ein Buchstabe). Dies ist für das Betriebssystem identisch mit dem Auswählen mit der Maus, wie man im Demo-Programm sehen kann. Die im Demoprogramm mitverwendete Definitionsdatei »Intuition.h« erlaubt nur einfach verzweigte Menü-Strukturen, das heißt, ein Menüunterpunkt kann keine weiteren Unterpunkte mehr haben. »Intuition« unterscheidet zwischen Aktions- und Attribut-Menüpunkten (Items). Attribut-Menüpunkte sind jene, welche, sofern angewählt, über das Aktivierungszeichen (Checkmark) verfügen. Ein so gewählter Menüpunkt kann nicht durch einfaches Anklicken wieder deaktiviert werden, sondern nur durch die »Mutual Exclusion«, der in Struktur »MenuItem« als »long integer« definiert ist. Dabei repräsentiert jeder Menüpunkt eines Menüs ein Bit, das erste Bit Null, das zweite Bit Eins, usw. In »MutualExclude« müssen deshalb die Bits jener Menüpunkte gesetzt werden, die nicht gleichzeitig aktiviert sein dürfen. Ein Beispiel dazu sehen Sie auch im Demo-Programm. Die für das Programm nötige Konfiguration sehen Sie noch einmal in Tabelle 3 zusammengestellt. Hier nun die Übergabeparameter für die Funktionen der Menüroutinen:

cr_MenuTitle (a,b,c)
- a: Zeiger auf ersten Menüpunkt
- b: Text des Titels
- c: Zeiger auf Menüstruktur

cr_MenuText (a,b,c,d,e)
- a: Zeiger auf nächsten Menüpunkt
- b: Text
- c: Shortcut (ASCII oder Null)
- d: Zeiger auf Menüpunkt-Struktur
- e: Zeiger auf IntuiText-Struktur

set_Attribute (a,b,c)
- a: Zeiger auf Menüpunkt
- b: MutualExclude
- c: Status (Aktivierungszeichen gesetzt = 1, gelöscht = 0)

cr_Menu (a,b,c,d,e,f,g)
- a: Zeiger auf das Window, zu dem dieses Menü gehört
- b: Zeiger auf alte Menüliste
- c: Zeiger auf anzuhängendes Menü
- d: X-Wert, an dem der Titel beginnt
- e: x-Wert, an dem die Menü-Punkte beginnen (relativ zu d)
- f: 1 = alle Boxen gleich lang (so lang wie die längste)
  0 = jede Box hat die Länge ihres Textes
- g: 1 = es kommen Attribute vor
  0 = nur Aktions-Menüpunkte

ce_Submenu (a,b,c,d,e,f)
- a: Zeiger auf den Menüpunkt, an den das Untermenü angehängt wird
- b: Zeiger auf den ersten Untermenüpunkt
- c: X-Wert, an dem die Untermenüs beginnen (relativ zu den Oberpunkten)
- d: 1 = Untermenüpunkte von aufrufendem Menüpunkt auch aufwärts
  0 = nur abwärts angeordnet
- e: 1 = alle Boxen gleich lang
  0 = jede Box hat die Länge ihres Textes
- f: 1 = Es kommen Attribute vor
  0 = Nur Aktions-Menüpunkte

ins_Line (a,b,c,d)
- a: Zeiger auf Menüpunkt, an dem der Text eingefügt werden soll
- b: Zeiger auf IntuiText-Struktur
- c: Auszugebender Text
- d: 1 = Hauptmenüpunkt
  0 = Untermenüpunkt

(Thomi Mauch/rf)

Programmname:	Menu.h
Sprache:	C
Compiler:	Lattice C V 3.03

Tabelle 3. Steckbrief für Menu.h

```
1   /********************************************/
2   /*** ProgrammName  : FType V2.2          ***/
3   /*** ProgrammAutor : Michael Rippl       ***/
4   /***                 Ringstr. 168        ***/
5   /***                 6096 Raunheim       ***/
6   /***                 Telefon : 06142/43617 ***/
7   /*** Computer      : AMIGA 1000          ***/
8   /*** Compiler      : Lattice C V3.03     ***/
9   /*** Datum         : 05-Jan-1986         ***/
10  /********************************************/
11
12  #include <exec/types.h>
13  #include <intuition/intuition.h>
14  #include <devices/printer.h>
15  #include <lattice/stdio.h>
16
17  #define MINPAR 2
18  #define MAXPAR 11
```

Listing 1. Der Quellcode zu Ftype.c

```
19   #define MAXCOLS 260
20
21   union printerIO  {
22      struct IOStdReq ios;
23      struct IODRPReq iodrp;
24      struct IOPrtCmdReq iopc;
25   };
26
27   union printerIO *request;
28
29   extern struct IORequest *CreateExtIO();
30   extern struct MsgPort *CreatePort();
31
32   /*** Hauptprogramm ***/
33   main(argc, argv)
34   int argc;
35   char *argv[];
36   {
37      struct MsgPort *printerPort;
38      FILE *FPtr;
39      BOOL Page, Numbering, LineFeed, LRMargins;
40      int Error, NrOfLines, Lines, NrOfPage, CMD_Nr[6], i, CMD_Anzahl,
                                                Length, DeutscheNorm;
41      char String[MAXCOLS], NString[MAXCOLS + 14];
42
43      Page = Numbering = LineFeed = LRMargins = FALSE;
44      CMD_Anzahl = -1;
45      DeutscheNorm = 72;
46
47   /*** Uebergebene Parameter werden ueberprueft ***/
48      if (argc >= MINPAR)  {
49         if (*argv[1] == '?')  {
50            printf(" Copyright 1986 By Michael Rippl\n");
51            printf("                   Telefon : 06142/43617\n\n");
52            printf("USAGE : ftype NAME ZEILENANZAHL -pnml cmd_1 cmd_2
                                          cmd_3 cmd_4 cmd_5 cmd_6\n");
53            exit(TRUE);
54         }
55      }
56      if ((argc <= MINPAR) || (argc >= MAXPAR))  {
57         printf("Bad Args\n");
58         exit(FALSE);
59      }
60      if (argc >= 4)  {
61         if (*argv[3] == '-')  {
62            Length = strlen(argv[3]) - 1;
63            for (i = 1; i <= Length; ++i)  {
64               if (*(argv[3] + i) == 'p')  {
65                  Page = TRUE;
66                  DeutscheNorm -= 2;
67               }
68               if (*(argv[3] + i) == 'n') Numbering = TRUE;
69               if (*(argv[3] + i) == 'm') LRMargins = TRUE;
70               if (*(argv[3] + i) == 'l') LineFeed = TRUE;
71            }
72            if ((Length == 0) || (Length > 4))  {
73               printf("Bad Args\n");
74               exit(FALSE);
75            }
76            for (i = 0; i <= (argc -5); ++i) CMD_Nr[i] = atoi(argv[i +
77            CMD_Anzahl = argc - 5;                                  4]);
78         }
79         if (*argv[3] != '-')  {
80            if (argc >= (MAXPAR - 1))  {
81               printf("Bad Args\n");
82               exit(FALSE);
83            }
84            for (i = 0; i <= (argc - 4); ++i) CMD_Nr[i] = atoi(argv[i
                                                                   + 3]);
85            CMD_Anzahl = argc - 4;
86         }
87      }
88      if (((NrOfLines = atoi(argv[2])) < 1) || ((LRMargins == TRUE) &&
                                             (CMD_Anzahl < 1)))  {
89         printf("Bad Args\n");
90         exit(FALSE);
91      }
92
93   /*** Textfile wird eroeffnet ***/
94      if((FPtr = fopen(argv[1], "r")) == NULL)  {
95         printf("Can't open %s\n", argv[1]);
96         exit(FALSE);
97      }
98
99   /*** Druckerinitialisierung ***/
100     printerPort = CreatePort("my.print.port", 0);
101     request = (union printerIO *)CreateExtIO(printerPort, sizeof
        (union
                                                          printerIO));
102
103     if ((Error = OpenPrinter(request)) != NULL)  {
104        DeleteExtIO(request, sizeof(union printerIO));
105        DeletePort(printerPort);
106        printf("Can't open prt:\n");
107        exit(FALSE);
108     }
109
110  /*** Nullbyte senden, damit neue Parameter akzeptiert werden ***/
111     PrintString(request, "\x00");
112
113  /*** CMD_Nummern an Drucker senden ***/
114     if (LRMargins == TRUE)  {
115        for(i = 2; i <= CMD_Anzahl; ++i) PrintCommand(request,
                                        CMD_Nr[i], 0, 0, 0, 0);
116        PrintCommand(request, aSLRM, CMD_Nr[0], CMD_Nr[1], 0, 0);
117     }
118     else for(i = 0; i <= CMD_Anzahl; ++i)  {
119        PrintCommand(request, CMD_Nr[i], 0, 0, 0, 0);
120        if (CMD_Nr[i] == 55) DeutscheNorm += 24;
121     }
122
123     Lines = 0;
124     NrOfPage = 1;
125
126  /*** Seitennummerierung fuer erste Druckerseite senden ***/
127     if (Page == TRUE)  {
128        NrOfLines -= 2;
129        sprintf(String, "Page : %d\n\n", NrOfPage);
130        PrintCommand(request, aSGR4, 0, 0, 0, 0);
131        PrintString(request, String);
132        PrintCommand(request, aSGR24, 0, 0, 0, 0);
133     }
134
135  /*** Textfile ausgeben ***/
136     while ((Error = fgets(String, (MAXCOLS - 2), FPtr) != NULL))  {
137        ++Lines;
138        if (Numbering == TRUE)  {
139           sprintf(NString, "%6d  %s", Lines, String);
140           PrintString(request, NString);
141        }
142        else PrintString(request, String);
143        if ((Lines % NrOfLines) == NULL)  {
144           ++NrOfPage;
145           if (LineFeed == TRUE) for (i = 0; i < (DeutscheNorm -
                 NrOfLines); ++i) PrintString(request, "\n\000");
146           else PrintString(request, "\f\000");
147           if (Page == TRUE)  {
148              sprintf(String, "Page : %d\n\n", NrOfPage);
149              PrintCommand(request, aSGR4, 0, 0, 0, 0);
150              PrintString(request, String);
151              PrintCommand(request, aSGR24, 0, 0, 0, 0);
152           }
153        }
154     }
155
156  /*** Files und Druckerport schliessen ***/
157     fclose(FPtr);
158     ClosePrinter(request);
159     DeleteExtIO(request, sizeof(union printerIO));
160     DeletePort(printerPort);
161  }
162
163  /*** Unterprogramme ***/
164  int OpenPrinter(request)
165  union printerIO *request;
166  {
167     return(OpenDevice("printer.device", 0, request, 0));
168  }
169
170  int ClosePrinter(request)
171  union printerIO *request;
172  {
173     return(CloseDevice(request));
174  }
175
176  /*** Textstring zum Drucker senden ***/
177  int PrintString(request, string)
178  union printerIO *request;
179  char *string;
180  {
181     request -> ios.io_Command = CMD_WRITE;
182     request -> ios.io_Data = (APTR)string;
183     request -> ios.io_Length = -1;
184     return(DoIO(request));
185  }
186
187  /*** Kommando zum Drucker senden ***/
188  int PrintCommand(request, command, p0, p1, p2, p3)
189  union printerIO *request;
190  int command, p0, p1, p2, p3;
191  {
192     request -> iopc.io_Command = PRD_PRTCOMMAND;
193     request -> iopc.io_PrtCommand = command;
194     request -> iopc.io_Parm0 = p0;
195     request -> iopc.io_Parm1 = p1;
196     request -> iopc.io_Parm2 = p2;
197     request -> iopc.io_Parm3 = p3;
198     return(DoIO(request));
199  }
```

Listing 1. Der Quellcode zu Ftype.c (Schluß)

```
1      /*****************************/
2      /* InitCLI  Version 1.1  03/87 */
3      /* von  Torsten Jürgleit       */
4      /* (c) Markt & Technik 1987    */
5      /*****************************/
6
7
8      #include <exec/types.h>
9      #include <exec/memory.h>
10     #include <graphics/gfxbase.h>
11     #include <graphics/display.h>
12     #include <graphics/text.h>
13     #include <intuition/intuition.h>
14     #include <functions.h>
15
16     #define VERSION    "1.1"
17     #define REV        0L
18     #define PSIZE      200L
19     #define MAXFLAGS   8
20     #define NAMESIZE   40
21
22     GLOBAL PLANEPTR AllocRaster();
23
24     struct GfxBase         *GfxBase = NULL;
25     struct ViewPort        *vp;
26     struct ColorMap        *cm;
27     struct BitMap          *b;
28     struct IntuitionBase   *IntuitionBase = NULL;
29     struct Window          *w;
30
31     BYTE  *flags[MAXFLAGS] = {"?","-s","-S","-c","-C","-w","-i","-I"};
32     UWORD mono_colors[] = {0x000,0x0f0,0x000,0x000};
33     UWORD oldcolors[3];
34
35                 /* Haupt-Programm */
36     VOID
37     main(argc,argv)
38        REGISTER argc;
39        BYTE     *argv[];
40     {
41        REGISTER i;
42        BYTE     flag[2], name[NAMESIZE];
43
44        if(!(GfxBase = (struct GfxBase *)
45                  OpenLibrary("graphics.library",REV))) {
46           printf("Kann 'graphics.library' nicht oeffnen !\n\n");
47           exit_program(1);
48        }
49        if(!(IntuitionBase = (struct IntuitionBase *)
50                  OpenLibrary("intuition.library",REV))) {
51           printf("Kann 'intuition.library' nicht oeffnen !\n\n");
52           exit_program(2);
53        }
54
55        vp = GfxBase->ActiView->ViewPort;
56        b  = vp->RasInfo->BitMap;
57        w  = IntuitionBase->ActiveWindow;
58
59        if(argc == 1) {
60           printf("Flags fehlen !\n\n");
61           exit_program(3);
62        }
63        for(i=1; i<argc; i++) {
64           strncpy(flag,argv[i],2);
65           switch(check_flags(flag)) {
66              case -1 :
67                 printf("Falsche Flags !\n\n");
68                 exit_program(4);
69                 break;
70              case 0 :
71                 print_usage();
72                 break;
73              case 1 :
74                 solobitmap_off();
75                 break;
76              case 2 :
77                 solobitmap_on();
78                 break;
79              case 3 :
80                 default_colors();
81                 break;
82              case 4 :
83                 monochrom();
84                 break;
85              case 5 :
86                 max_window();
87                 break;
88              case 6 :
89                 interlace_off();
90                 break;
91              case 7 :
92                 interlace_on();
93                 break;
94           }
95        }
96        exit_program(0);
```

Listing 2. INITCLI.C — Utility für das CLI des Amiga

```
97   }
98                  /* Bedienungshinweise */
99
100  print_usage()
101  {
102     printf("InitCLI: Version %s 03/87 von Torsten
                         Juergeleit\n\n",VERSION);
103     printf(" Eingabeformat : InitCLI [?][-s][-S][-c][-C][-w][-i][
                                                       -I]\n");
104     printf(" Bedeutung der Flags :\n");
105     printf("      ?              = Gibt diese Hilfsinformation
                                                       aus\n");
106     printf("      -s             = SoloBitmap ausschalten\n");
107     printf("      -S             = SoloBitmap einschalten\n");
108     printf("      -c             = Farben aus Preferences
                                          einschalten\n");
109     printf("      -C             = Monochrom-Modus
                                      einschalten\n");
110     printf("      -w             = Maximale Window-Groesse\n");
111     printf("      -i             = Interlace Modus
                                      ausschalten\n");
112     printf("      -I             = Interlace Modus
                                    einschalten\n\n");
113     exit_program(0);
114  }
115                 /* Programm verlassen */
116
117  exit_program(error)
118     SHORT error;
119  {
120     if(GfxBase)         CloseLibrary(GfxBase);
121     if(IntuitionBase)   CloseLibrary(IntuitionBase);
122     exit(0);
123  }
124                 /* Flags ueberpruefen */
125
126  check_flags(flag)
127     BYTE flag[2];
128  {
129     REGISTER i;
130
131     for(i=0; i<MAXFLAGS; i++) {
132        if(!(strcmp(flag,flags[i]))) {
133           return(i);
134        }
135     }
136     return(-1);
137  }
138             /* Zweite Bitplane ausschalten */
139
140  solobitmap_on()
141  {
142     if(b->Depth == 2) {
143        b->Depth = 1;
144        FreeRaster(b->Planes[1],640L,(ULONG)vp->DHeight);
145        RemakeDisplay();
146     }
147  }
148               /* Zweite Bitplane einschalten */
149
150  solobitmap_off()
151  {
152     PLANEPTR planeptr;
153
154     if(b->Depth == 1) {
155        if((planeptr = AllocRaster(640L,(ULONG)vp->DHeight)) !=
                                                        NULL) {
156           b->Depth     = 2;
157           b->Planes[1] = planeptr;
158           BltClear(planeptr,(ULONG)RASSIZE(640,vp->DHeight),0L);
159           RemakeDisplay();
160        }
161     }
162  }
163               /* Monochrom-Modus einschalten */
164
165  monochrom()
166  {
167     FreeColorMap(vp->ColorMap);
168     vp->ColorMap = GetColorMap((LONG)(2 * b->Depth));
169     LoadRGB4(vp,mono_colors,(LONG)(2 * b->Depth));
170  }
171  /* Urspruengliche Farben einschalten */
172
173  default_colors()
174  {
175     struct Preferences *p;
176
177     if((p = GetPrefs((struct Preferences *)
178     AllocMem(PSIZE,MEMF_CLEAR),PSIZE)) != NULL{
179        oldcolors[0] = p->color0;
180        oldcolors[1] = p->color1;
181        oldcolors[2] = p->color2;
182        oldcolors[3] = p->color3;
183        FreeMem(p,PSIZE);
184        FreeColorMap(vp->ColorMap);
185        vp->ColorMap = GetColorMap((LONG)(4 * b->Depth));
186        LoadRGB4(vp,oldcolors,(LONG)(4 * b->Depth));
187     }
188  }
189        /* Window auf Maximalgroesse bringen */
190
191  max_window()
192  {
193     ULONG deltax,deltay;
194
195     if(w != NULL) {
196        deltax = - (ULONG)w->LeftEdge;
197        deltay = - (ULONG)w->TopEdge;
198        MoveWindow(w,deltax,deltay);
199        deltax = (ULONG)IntuitionBase->ActiveScreen->Width -
                                              (ULONG)w->Width;
200        deltay = (ULONG)IntuitionBase->ActiveScreen->Height -
                                              (ULONG)w->Height;
201        SizeWindow(w,deltax,deltay);
202     }
203  }
204         /* Interlace Modus einschalten */
205
206  interlace_on()
207  {
208     GfxBase->system_bplcon0 |= INTERLACE;
209     RemakeDisplay();
210  }
211         /* Interlace Modus ausschalten */
212
213  interlace_off()
214  {
215     GfxBase->system_bplcon0 &= !INTERLACE;
216     RemakeDisplay();
217  }
```

Listing 2. (Schluß)

```
1   #include "df1:lenght.h"
2   /************************************************/
3   /*       Menu-Routinen für Amiga                */
4   /*       (C) 1987 Markt & Technik               */
5   /*       by Thomi Mauch                         */
6   /*       Augustin Keller-Str. 9                 */
7   /*       CH-5000 Aarau                          */
8   /************************************************/
9   /*  Werden die folgenden Routinen im LoRes-Modus  */
10  /*  (320*200 / 320*400) verwendet, so sollten     */
11  /*     CHECKWIDTH und    COMMWIDTH durch          */
12  /*  LOWCHECKWIDTH und LOWCOMMWIDTH ersetzt werden */
13  /************************************************/
14  cr_MenuTitle (p,text,new)          /* Teilweises Initiali- */
15  struct MenuItem *p;                /* sieren des Menu-     */
16  BYTE *text;                        /* Headers              */
17  struct Menu *new;
18  {
19     new->TopEdge   = 0;
20     new->Width     = lenght (text) * 8 + 8;
21     new->Height    = 0;
22     new->Flags     = MENUENABLED | MIDRAWN;/* Flags für Auswählen*/
23     new->MenuName  = text;                  /* möglich und zum    */
24     new->FirstItem = p;                     /* Zeichnen auf den   */
25  }                                          /* Bildschirm         */
26
27  cr_MenuText (p,text,command,new1,new2)   /* Teilweises Initiali-*/
28  struct MenuItem *p;                       /* sieren eines Menu-  */
29  BYTE *text;                               /* Items und des dazu- */
30  BYTE command;                             /* gehörenden Intui-   */
31  struct MenuItem  *new1;                   /* Textes              */
32  struct IntuiText *new2;
33  {
34     new2->FrontPen  = 0;                   /* Init IntuiText      */
35     new2->BackPen   = 0;
36     new2->DrawMode  = JAM1;
37     new2->LeftEdge  = 0;
38     new2->TopEdge   = 1;
39     new2->ITextFont = NULL;
40     new2->IText     = text;
41     new2->NextText  = NULL;
42
43     new1->NextItem      = p;               /* Init MenuItem       */
44     new1->Width         = lenght (text) * 8;
45     new1->Height        = 10;
46     new1->Flags         = ITEMTEXT    |    /* Flags fuer Text,    */
47                           ITEMENABLED |    /* Auswählen möglich,  */
48                           HIGHCOMP    |    /* Revers-Darstellung  */
49            (COMMSEQ * (command>1));        /* und Shortcut        */
50     new1->MutualExclude = 0;
51     new1->ItemFill      = (APTR) new2;     /* Pointer auf Text    */
```

Listing 3. Die Menüroutinen »menu.h« für den Einsatz in C-Programmen

Ihr Spezialist für AMIGA-Software

Zur Zeit ca. 200 Software-Angebote (ohne Public Domain) verfügbar!

Aus unserem Programm-Angebot mit deutschen Handbüchern:

Key to »C« — Programmierutilities für C-Programmierer, unterstützt *AMIGA*-spezifische Befehle — **148,–***

A-Mon — Disketten- und Programm-Monitor. Digitalisierter Sound kann abgehört werden. Sprites lassen sich editieren! *AMIGA-ARTISTS* — **138,–***

BOOT BOY — Grafiktitel kann in Bootblock eingebunden werden! Nur eine Sekunde nach Einlegen der Diskette erscheint dieser auf dem Monitor - *AMIGA-ARTISTS* — **69,–***

Diskwik — Diskettenmonitor. Es kann in Hex und ASCII editiert werden — **108,–***

SUPERWRITER — IFF-kompatibler Titelmaker, auch digitalisierte Musik möglich - *AMIGA-ARTISTS* — **99,–***

Neuvorstellungen aus USA und Deutschland:

SONIX-DREAMS — Musikstücke und Instrumente. Zubehördiskette zu dem Musikprogramm ***AEGIS SONIX.*** Von *ANDREAS STARR.* Inzwischen sind schon Nr. 1 und 2 erhältlich! — je **58,–***

AMIGA COLORING BOOK — Malbuch für IFF-kompatible Malprogramme! Vorlagen können in eigene Bilder eingearbeitet werden. Nr. 1-3 erhältlich. — je **88,–***

DIGI-PIX — Digitalisierte Bilder im IFF-Format — **78,–***

DARÜBER HINAUS BIETEN WIR FAST ALLE AUF DEM WELTMARKT ERHÄLTLICHEN SOFTWARE-PRODUKTE ZU SUPERPREISEN AN:

Produkt	Preis	Produkt	Preis
Aegis Draw Plus	508,-*	**DigiView**	448,-*
Aegis Sonix: Musikprogramm	178,-*	**Flight Simulator 2**	118,-*
Autoduell	118,-*	**Future Sound**	398,-*
Deluxe Paint 2	278,-*	**Jet**	118,-*
Deluxe Paint 2 mit dt. Anwenderbuch	303,-*	**Page Setter**	328,-*

Disketten 2DD (135 tpi) von 2,45 - 2,90 DM pro Stück

KOSTENLOSE INFO UND PRODUKTLISTE ANFORDERN!

* Alle Preise sind unverbindliche Preisempfehlungen!



Weiterhin sucht ATLANTIS fähige Programmierer aus der AMIGA-Szene für alle Anwendungsgebiete! Falls Sie Interesse haben, setzen Sie sich mit uns in Verbindung.

ATLANTIS arbeitet eng mit den Commodore-Systemhändlern zusammen.

Vertriebsgesellschaft
In der Henn 27, 5030 Hürth
☎ (02233) 35735

```
52      new1->SelectFill    = (APTR) new2;
53      new1->Command       = command;        /* Shortcut          */
54      new1->SubItem       = NULL;
55      new1->NextSelect    = 0;
56   }
57
58   set_Attribute (item,exclude,state)        /* Kennzeichnet Menu- */
59   struct MenuItem *item;                    /* Item als Attribut  */
60   LONG exclude;                             /* und setzt CHECKED  */
61   int state;                                /* und MutualExclude  */
62   {
63      item->Flags        = item->Flags | CHECKIT | (state > 0) *
                                                     CHECKED;
64      item->MutualExclude = exclude;
65   }
66
67   cr_Menu (window,old,new,x,xItem,mode,attr) /* Initialisiert Menu*/
68   struct Window *window;                     /* Header und alle   */
69   struct Menu *old,*new;                     /* MenuItems fertig, */
70   int x,xItem,mode,attr;                     /* sorgt für die gra-*/
71   {                                          /* fische Gestaltung */
72      int down = 10;                          /* und initialisiert */
73      int highest;                            /* auf dem Bildschirm*/
74      struct MenuItem *item1,*item2;
75
76      new->NextMenu = old;                    /* MenuHeader fertig */
77      new->LeftEdge = x;                      /* initialisieren    */
78
79      item1 = new->FirstItem;                 /* Erstes MenuItem   */
80      item1->LeftEdge = xItem;                /* behandeln         */
81      item1->TopEdge  = 0;
82      if (attr)
83         attribute (item1);
84      highest = item1->Width;
85
86      while ((item2 = item1->NextItem) != NULL) /* Schleife für */
87         {                                       /* alle anderen */
88         item2->LeftEdge = xItem;
89         item2->TopEdge  = down;
90         if (attr)
91            attribute (item2);
92         if (item2->Width > highest)
93            highest = item2->Width;
94         item1 = item2;
95         down += 10;
96         }
97
98      if (mode)                          /* Grafische Gestaltung */
99         equal (new->FirstItem,highest); /* Boxen gleich lang    */
100     shortcut (new->FirstItem,highest); /* Platz fuer Shortcuts */
101
102     ClearMenuStrip (window);           /* Neue Menu-Liste      */
103     SetMenuStrip   (window,new);       /* zeigen               */
104  }
105
106  cr_SubMenu (pItem,pSub,x,yMode,mode,attr) /* Initialisiert alle */
107  struct MenuItem *pItem,*pSub;             /* MenuItems fertig und*/
108  int x,yMode,mode,attr;                    /* sorgt für die gra-  */
109  {                                         /* fische Gestaltung   */
110     struct MenuItem *item1,*item2;
111     int number = 1;
112     int highest;
113     int down;
114
115     pItem->SubItem = pSub;                 /* Erstes MenuItem  */
116     item1 = pSub;                          /* behandeln        */
117     if (attr)
118        attribute (item1);
119     highest = item1->Width;
120     item1->LeftEdge = x;
121     ((struct IntuiText *)item1->ItemFill)->TopEdge  = 0;
122     item1->Height   = 8;
123
124     while ((item2 = item1->NextItem) != NULL) /*Schleife für alle*/
125        {                                      /* anderen         */
126        number++;
127        if (attr)
128           attribute (item2);
129        if (item2->Width > highest)
130           highest = item2->Width;
131        item2->LeftEdge = x;
132        ((struct IntuiText *)item2->ItemFill)->TopEdge  = 0;
133        item2->Height   = 8;
134        item1 = item2;
135        }
136
137     if (mode)                          /* Grafische Gestaltung */
138        equal (pSub,highest);           /* Boxen gleich lang    */
139     shortcut (pSub,highest);           /* Platz fuer Shortcuts */
140
141     if (yMode == 0)                    /* SubItems nur ab-     */
142        down = 0;                       /* wärts oder zen-      */
143     else                               /* triert               */
144        down = 0 - (number-1>>1) * 8;
145
146     item1 = pSub;
147     item1->TopEdge = down;
148     while ((item2 = item1->NextItem) != NULL)
149        {
150        down += 8;
151        item2->TopEdge = down;
152        item1 = item2;
153        }
154  }
155
156  shortcut (item1,highest)                 /* Wenn Shortcuts, dann */
157  struct MenuItem *item1;                  /* Shortcuts an rechten */
158  int highest;                             /* Rand.                */
159  {
160     struct MenuItem *item2;
161
162     if (item1->Flags & COMMSEQ)
163        item1->Width = highest + COMMWIDTH + 16;
164     while ((item2 = item1->NextItem) != NULL)
165        {
166        if (item2->Flags & COMMSEQ)
167           item2->Width = highest + COMMWIDTH + 16;
168        item1 = item2;
169        }
170  }
171
172  equal (item1,highest)                    /* Alle Selectboxes     */
173  struct MenuItem *item1;                  /* gleich lang machen.  */
174  int highest;
175  {
176     struct MenuItem *item2;
177
178     item1->Width = highest;
179     while ((item2 = item1->NextItem) != NULL)
180        {
181        item2->Width = highest;
182        item1 = item2;
183        }
184  }
185
186  attribute (item)                         /* Wenn Attribute, dann */
187  struct MenuItem *item;                   /* Attribute an linken  */
188  {                                        /* Rand                 */
189     item->Width += CHECKWIDTH;
190     ((struct IntuiText *)item->ItemFill)->LeftEdge = CHECKWIDTH;
191  }
192
193  ins_Line (item,text,string,mode)         /* Abstand zwischen     */
194  struct MenuItem  *item;                  /* zwei MenuItems er-   */
195  struct IntuiText *text;                  /* zeugen und dort ei-  */
196  UBYTE            *string;                /* nen String ausgeben  */
197  USHORT           mode;
198  {
199     struct MenuItem *item1,*item2;
200
201     if (mode == 1)                        /* Unterschiedliche     */
202        mode = 8;                          /* Grösse Item / Sub-   */
203     else                                  /* Item berücksichtigen */
204        mode = 6;
205     item1 = item;
206     while ((item2 = item1->NextItem) != NULL) /* Schleife: Items */
207        {                                      /* nach unten      */
208        item2->TopEdge += mode;                /* schieben        */
209        item1 = item2;
210        }
211
212     ((struct IntuiText *)item->ItemFill)->NextText  = text;
213     text->FrontPen  = 0;
214     text->BackPen   = 0;                  /* IntuiText-Struktur   */
215     text->DrawMode  = JAM1;               /* initialisieren und   */
216     text->LeftEdge  = 0;                  /* anhängen             */
217     text->TopEdge   = mode + (mode==8) + 1;
218     text->ITextFont = NULL;
219     text->IText     = string;
220     text->NextText  = NULL;
221  }
222
```

Listing 3. (Schluß)

```
1  lenght (string)
2  char string[];
3  {
4        int i=0;
5        while (string[i++] != '\0');
6        return(--i)
7  }
```

Listing 4. Die Funktion »lenght«, die von »menu.h« benötigt wird

```
1   /*********************************/
2   /***     Menu-Demo für Amiga    ***/
3   /***        (C) 1987            ***/
4   /***     Markt & Technik        ***/
5   /***     by Thomi Mauch         ***/
6   /***     Augustin Keller-Str. 9 ***/
7   /***     CH-5000 Aarau          ***/
8   /*********************************/
9
10  #include "exec/types.h"
11  #include "intuition/intuition.h"
12  #include "df1:menu.h"
13
14  struct Window *OpenWindow ();
15  struct Screen *OpenScreen ();
16
17  long   GfxBase       = 0;
18  long   IntuitionBase = 0;
19  USHORT class;
20  USHORT code;
21  struct Window        *w;
22  struct Screen        *screen;
23  struct RastPort      *rp;
24  struct IntuiMessage *message;
25
26  /* Wenn Kickstart 1.2 und Workbench 1.2 benutzt werden, kann   */
27  /* die Höhe des Screens selbstverständlich auf 256 erhöht      */
28  /*werden, ebenso die Höhe des Windows.  */
29
30  struct NewScreen ns = {
31     0,0,          /* x-, y-Startwert */
32     640,200,2,    /* Breite, Höhe, Anzahl BitPlanes */
33     0,1,          /* Vorder-, Hintergrundfarbe*/
34     HIRES,        /* Flags für Screen-Typ     */
35     CUSTOMSCREEN,/* Von Anwender definiert   */
36     NULL,         /* Zeiger auf Zeichensatz   */
37     "Menu Demo", /* Titel                    */
38     NULL          /* Zeiger auf Gadgets       */
39     };
40
41  struct NewWindow nw = {
42        40,20,                  /* x-, y-Startwert          */
43        560,160,                /* Breite, Höhe             */
44        0,1,                    /* Vorder-, Hintergrundfarbe*/
45     CLOSEWINDOW | MENUPICK, /* IDCMP-Flags   */
46     WINDOWCLOSE | ACTIVATE, /* Flags für Window-Typ*/
47        NULL,                   /* Zeiger auf Gadgets       */
48        NULL,                   /* Zeiger auf eigenes Checkmark*/
49        "Menu",                 /* Titel                    */
50        NULL,                   /* Zeiger auf Screen-Struktur*/
51        NULL,                   /* Zeiger auf BitMap        */
52        0,0,0,0,                /* Minimal-, Maximalgrösse */
53        CUSTOMSCREEN            /* Screen-Typ           */
54        };
55
56  /*Der Aufruf Text() gehrt zu den 'graphics primitive', d.h. */
57  /*es ist eine einfache Betriebssystem-Routine, die demzufolge */
58  /*auch keinen Komfort bietet. So ist Text () zum Beispiel     */
59  /*nicht in der Lage, Werte von Variablen auszugeben. Deshalb  */
60  /*werden die       nur einstelligen Zahlen mit Hilfe eines    */
61  /*char-Arrays in einen String überführt und diese dann aus-   */
62  /*gegeben.                                                    */
63
64  main ()
65  {
66     BOOL   cont = 1;
67     USHORT menunum;
68     USHORT itemnum;
69     USHORT subnum;
70  static char zahl [] = { '0','1','2','3','4','5','6','7','8','9' };
71     struct Menu        a,b,c;
72     struct MenuItem    c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,
73                        ca,cb,cc,cd,ce,cf,cg,ch,ci,cj;
74     struct IntuiText   d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,da,
75                        db,dc,dd,de,df,dg,dh,di,dj,dk,dl;
76
77     GfxBase = OpenLibrary("graphics.library",0);
78     if(GfxBase == NULL)
79        exit();
80
81     IntuitionBase = OpenLibrary("intuition.library",0);
82     if(IntuitionBase == NULL)
83        exit();
84
85     screen = OpenScreen(&ns);
86     if(screen == NULL)
87        exit(0);
88
89     nw.Screen = screen;          /* Wichtig: In der NewWindow-   */
90     w         = OpenWindow(&nw); /* Struktur wird der Pointer auf*/
91     rp        = w->RPort;        /* den Screen, nicht auf die    */
92                                  /* NewScreen-Struktur erwartet! */
93     cr_MenuTitle  (&c0,"Datei",&a);
94     cr_MenuText   (&c1,"Laden",0,&c0,&d0);
95     cr_MenuText   (NULL,"Drucken",0,&c1,&d1);
96     cr_Menu       (w,NULL,&a,20,-10,1,0);
97     cr_MenuText   (&c3,"bernehmen",0,&c2,&d2);
98     cr_MenuText   (NULL,"-ndern",0,&c3,&d3);
99     cr_SubMenu    (&c0,&c2,50,0,0,0);
100    cr_MenuText   (&c5,"Schnschrift",0,&c4,&d4);
101    cr_MenuText   (&c6,"Schnellschrift",0,&c5,&d5);
102    cr_MenuText   (&c7,"Endlos-Papier",0,&c6,&d6);
103    cr_MenuText   (NULL,"Einzelblatt",0,&c7,&d7);
104    set_Attribute (&c6,0xfffb,1);
105    set_Attribute (&c7,0xfff7,0);
106    cr_SubMenu    (&c1,&c4,50,1,0,1);
107    ins_Line      (&c0,&d1,"-------",1);
108    ins_Line      (&c5,&dk,"----------------",0);
109
110    cr_MenuTitle  (&c8,"Tastatur",&b);
111    cr_MenuText   (&c9,"Deutsch",'D',&c8,&d8);
112    cr_MenuText   (&ca,"Francais",'F',&c9,&d9);
113    cr_MenuText   (NULL,"English",'E',&ca,&da);
114    set_Attribute (&c8,0xfffe,1);
115    set_Attribute (&c9,0xfffd,0);
116    set_Attribute (&ca,0xfffb,0);
117    cr_Menu       (w,&a,&b,130,0,0,1);
118
119    cr_MenuTitle  (&cb,"Zeichen",&c);
120    cr_MenuText   (&cc,"Zeichensatz",0,&cb,&db);
121    cr_MenuText   (NULL,"Darstellung",0,&cc,&dc);
122    cr_Menu       (w,&b,&c,250,0,0,0);
123    cr_MenuText   (&ce,"Laden",0,&cd,&dd);
124    cr_MenuText   (&cf,"Drucken",0,&ce,&de);
125    cr_MenuText   (NULL,"Ändern",0,&cf,&df);
126    cr_SubMenu    (&cb,&cd,70,0,1,0);
127    cr_MenuText   (&ch,"Normal",'N',&cg,&dg);
128    cr_MenuText   (&ci,"Bold",'B',&ch,&dh);
129    cr_MenuText   (&cj,"Underlined",'U',&ci,&di);
130    cr_MenuText   (NULL,"Italic",'I',&cj,&dj);
131    set_Attribute (&cg,0xfffe,1);
132    set_Attribute (&ch,0x0001,0);
133    set_Attribute (&ci,0x0001,0);
134    set_Attribute (&cj,0x0001,0);
135    cr_SubMenu    (&cc,&cg,50,1,0,1);
136
137    while (cont)
138       {
139       if (w->UserPort->mp_SigBit)     /* Nachricht      */
140          {                            /* vorhanden ?    */
141          message = (struct IntuiMessage *)GetMsg(w->UserPort);
142          if (message != NULL)
143             {                         /* Wichtigste Da- */
144             class = message->Class;  /* ten speichern, */
145             code  = message->Code;   /* Nachricht wei- */
146             ReplyMsg (message);      /* tergeben       */
147
148             switch (class)
149                {
150                case CLOSEWINDOW:        /* Benutzer       */
151                   cont = 0;             /* schloss Fen-   */
152                   break;                /* ster -> Ende   */
153                case MENUPICK:           /* rechter Knopf: */
154                   SetAPen  (rp,0);     /* Bereich für    */
155                   RectFill (rp,20,50,400,70);/* Mitteilungen  */
156                   SetAPen  (rp,1);             /* löschen       */
157                   if (code != MENUNULL)        /* Menu angewählt */
158                      {
159                      menunum = MENUNUM (code);/* Menu, Item,     */
160                      itemnum = ITEMNUM (code);/* SubItem aus-    */
161                      subnum  = SUBNUM  (code);/* geben           */
162
163                      Move (rp, 20,57);  Text (rp,"Menu   Item
                                                  SubItem",21);
164                      Move (rp, 32,67);  Text (rp,&zahl
                                                  [menunum],1);
165                      Move (rp, 76,67);  Text (rp,&zahl
                                                  [itemnum],1);
166                      Move (rp,132,67);
167                      if (subnum != NOSUB)
168                         Text (rp,&zahl [subnum ],1);
169                      else
170                         Text (rp,"Kein SubItem",12);
171                      }
172                   else
173                      {
174                      Move (rp, 20,62);  Text (rp,"Kein Menu wurde
                                                  angewählt",25);
175                      }
176                   break;
177                }
178             }
179          }
180       }
181    CloseWindow (w);          /* Fenster zu */
182    CloseScreen (screen);     /* Screen zu */
183 }
184
```

Listing 5. Menudemo.c zeigt anschaulich die Leistung der Menüroutinen

Universeller Amiga-Monitor

Hier zeigen wir Ihnen, wie Sie den Amiga-Monitor 1081 problemlos mit perfekter Bildqualität an Videorecorder oder TV-Tuner anschließen können.

Wer sich bisher erfolgreich durch den Wald der unterschiedlichen Amiga-Ausführungen geschlagen hat, wird spätestens bei Inbetriebnahme des externen Monitor-Videoeinganges (CVBS) mit einem Videosignal vom Videorecorder oder von einem Fernseh-Tuner auf unlösbare Probleme stoßen.

Die Benutzung der externen Eingänge für Bild und Ton ist nur dann störungsfrei, wenn der Computer ausgeschaltet bleibt. Um diesen etwas merkwürdigen Zustand abzustellen, stellen wir Ihnen Schaltungsänderungen auf der Monitorplatine vor, die von jedem erfahrenen Bastler ohne Kostenaufwand durchzuführen sind.

Angefangen hat es eigentlich mit dem Kauf eines inzwischen recht preiswert erhältlichen Fernsehempfangsteiles, eines sogenannten Monitor-TV-Tuners, der das Antennensignal in Video- und Audiosignale umsetzt und über zwei Kabel, ausgestattet mit Cinchsteckern, den externen Eingängen CVBS IN und AUDIO IN des Amiga-Monitors vom Typ 1081 ohne Anschlußprobleme zuführt.

Die Bildqualität ist ausgesprochen gut, nicht zuletzt bedingt durch die recht kleine Pixelgröße des Amiga-Bildschirms.

Doch der Ärger kommt mit dem Einschalten des Computers.

Das bisher störungsfreie Bild wird unruhig, zittert und ist von deutlich sichtbaren Streifen durchzogen, die vom Amiga-Videosignal stammen.

In Schalterstellung »RGB« ist das Computerbild einwandfrei, doch hört man den Fernsehton in unverminderter Lautstärke, gemischt mit dem Tonsignal des Amiga (beide Kanäle).

Ein Blick in das mitgelieferte Schaltbild des Amiga-Monitors 1081 läßt den Grund für die Bildstörung recht schnell erkennen.

Unsynchronisiertes Signalgemisch

Über den Widerstand R 504 werden das externe Videosignal und das am Pin 20 des Scartsteckers anliegende Computersignal addiert. Dieses zueinander nicht synchrone Signalgemisch gelangt zu den weiterverarbeitenden Stufen, dem PAL-Demodulator und der Impulsabtrennstufe und verursacht dort Störungen in Synchronisation und Bildinhalt.

Mit dem Entfernen von R 504 ist der »Fehler« behoben, das heißt externes Videosignal und Computer-Bildsignal beeinflussen sich nicht mehr.

Natürlich darf nicht verschwiegen werden, daß der Monitor bei Betrieb zum Beispiel mit einem Videorecorder über Scart-Kabel kein Bild mehr zeigt, da der Zweig von der Scart-Buchse (Pin 20) zum Videoverstärker TS 508 unterbrochen wurde.

Wer den Scartanschluß braucht, kann R 504 über einen Miniaturschalter ein- und ausschalten. Andererseits kann ein Videorecorder aber auch über einen Scart-Cinch-Adapter (oder Kabel) an den Cinch-Eingang des Monitors angeschlossen werden.

Auch die Verknüpfung der Toneingänge ist aus dem Monitor-Schaltbild ersichtlich:

Das vom Amiga über den Scartstecker kommende Stereosignal (zwei Kanäle, Pin 2 und 6) wird an R 303 bis R 305 zusammengeschaltet (Monosignal) und über R 301, R 302, R 307 mit »AUDIO IN« verbunden. Das Gemisch erreicht schließlich über den Lautstärkeregler R 316 und den NF-Leistungsverstärker IC 301 den Monitorlautsprecher. Eine Umschaltung zwischen Computer-Ton und AUDIO IN erfolgt nicht.

Da der an der Frontplatte rechts neben dem Lautstärkeregler liegende Umschalter »RGB/CVBS« nur halbseitig für die Videosignalumschaltung genutzt wird, läßt sich durch eine einfache Schaltungsänderung auch der Ton mit umschalten, so daß in Stellung »RGB« Bild und Ton vom Amiga kommen und in Stellung »CVBS« vom Bild- und Toneingang.

Auch dieser Eingriff erfordert keinen Aufwand. Benötigt werden etwa 20 cm abgeschirmte Leitung und etwas Schaltdraht.

Die Verbindung des Lautstärkereglers R 316 mit den Widerständen R 307 und R 302 wird gelöst, R 316 nun mit dem RGB/CVBS-Umschaltermittenkontakt verbunden, R 302 mit dem linken Außenkontakt und R 307 entsprechend mit dem rechten.

Bild 1 zeigt die Originalschaltung im genannten Bereich und Bild 2 die Änderung.

Auch hier, mag der Leser einwenden, kann es in manchen Fällen hilfreich sein, zur Unterhaltung den externen Ton zu hören, zum Beispiel bei Bearbeitung einer umfassenden Kalkulation. Oder man möchte in der Zeit, in der der Amiga ein Programm abarbeitet, die Heute-Sendung im Fernsehen sehen oder einen spannenden Krimi verfolgen, aber dennoch nicht versäumen, das READY-Signal zu hören. Auch dieser Fall ist zu lösen, wenn man einen Schalter einbaut, der die beiden äußeren Kontakte bei Bedarf kurzschließt und damit den Originalzustand beim Ton wiederherstellt (gestrichelte Verbindung in Bild 2).

Bauanleitung:

Bevor man sich an die Arbeit macht und die Rückwand des

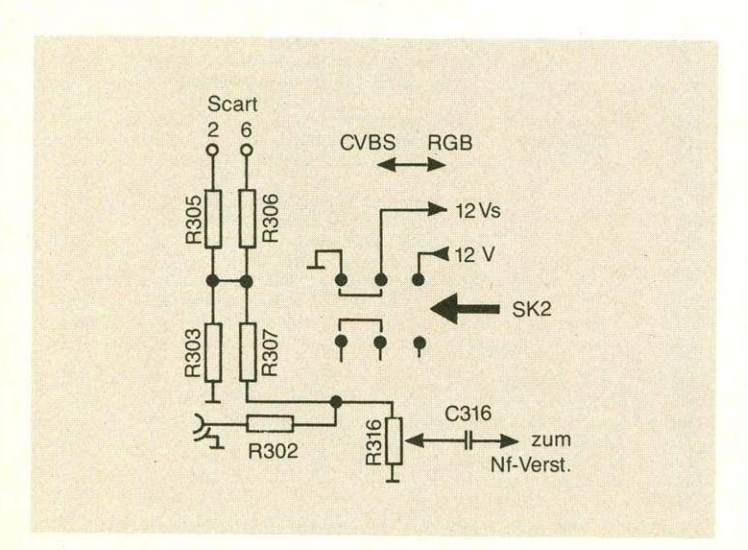


Bild 1. Auszug aus der Originalschaltung

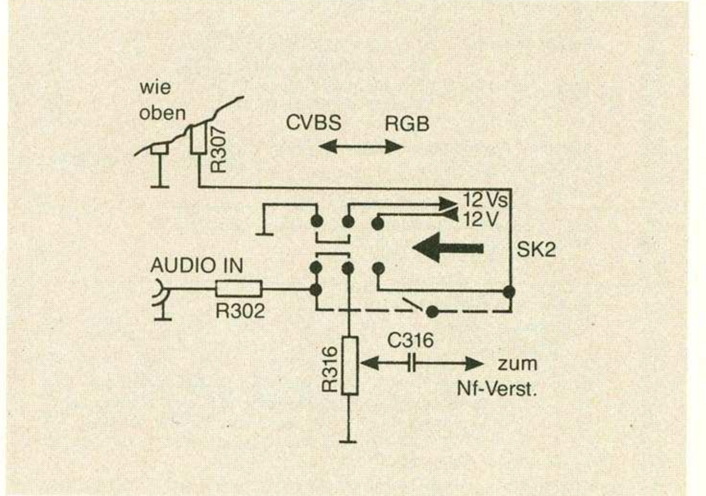


Bild 2. Geänderte Schaltung

Monitors löst, sollte man sich im klaren sein, daß das Arbeiten an Fernsehgeräten und damit auch an Monitoren nicht ungefährlich ist und wirklich dem Fachmann vorbehalten sein sollte. Wenn das Gerät ausgeschaltet wird, baut sich die Ladung an der Bildröhrenanode nur sehr langsam ab, so daß die verbleibende Spannung noch nach Tagen lebensgefährlich ist. Sie kann neben direkten Schäden (Stromschlag) Schreckensreaktionen auslösen, die zu mechanischen Beschädigungen bis hin zur Bildröhrenzerstörung mit ihren verheerenden Glassplittergeschossen führen können.

Daß man zuvor alle Kabel löst (Netzkabel!), um spannungsfrei arbeiten zu können, ist wohl selbstverständlich.

Doch nun zum Umbau:

Nach Lösen von je zwei Kreuzschlitzschrauben am oberen und unteren Gehäuseteil kann die Rückwand des Monitors entfernt werden. Auf der horizontal angeordneten Printplatte sind die Video- und die Tonverstärkerstufen angeordnet, an denen man die Änderungen vorzunehmen hat.

Video:

Unmittelbar hinter der Cinchbuchse »CVBS IN« (ganz links) befindet sich der Widerstand R 504, der einseitig oder ganz abzuwirken oder auszulöten ist. Wer sich auf diese Änderung beschränkt, der ist bereits fertig und kann das Gerät wieder schließen.

Ton:

Für die genannten Änderungen muß die Signalplatte herausgenommen werden. Hierzu sind alle Steckverbindungen behutsam zu lösen und zu markieren. Sodann läßt sich die Platine vorsichtig nach hinten herausziehen. Nun wird die über dem Lautstärkeregler R 316 und den Schiebeschaltern SK2 und SK4 liegende Abschirmplatte ausgelötet und auf die Seite gelegt. Der Innenleiter des am Poti befestigten weißen NF-Kabel wird abgelötet und am beschrifteten Lötpunkt 44 angelötet, die Abschirmung an Lötpunkt 45. Auf der Printseite wird eine Brücke (ein isolierter Schaltdraht) von C 308 (Minusseite), entspricht SK2 (Mitte), zum freien Punkt des Lautstärkereglers R 316 gelegt und in die freien Bohrlöcher eingelötet. Der Innenleiter eines etwa 20 cm langen NF-Kabels ist am Lötpunkt 50 anzulöten, die Abschirmung entsprechend an 49. Zur Befestigung des Kabelanfangs wird zunächst R 307 an der Seite ausgelötet, die R 501 am nächsten liegt. Das freigelegte Ende von R 307 wird hochgebogen und mit dem Innenleiter des Kabelanfangs zusammengelötet. Der Außenleiter ist an 21, an dem bereits der Außenleiter des weißen Kabels liegt, mit anzulöten.

Der 1081 in Stereo

Damit ist auch die Ton-Änderung erledigt. Die Abschirmplatte wird wieder eingelötet, die Platine eingesetzt, und mit allen Steckern verbunden. Nach Sichtkontrolle schraubt man die Rückwand wieder zu und verbindet den Monitor mit den zugehörigen Kabeln. Wenn sich kein Fehler eingeschlichen hat, funktioniert die Schaltung so, wie beschrieben.

Bei den Arbeiten fällt auf, daß die Signalplatte nur unvollständig, das heißt beschriftet, aber nicht mit Bauteilen bestückt ist. Der Schaltplan sagt darüber auch nichts aus. Im Schaltplan des früheren Monitors CM 8533 jedoch ist dem Lautstärkeregler R 316 eine Verstärkerstufe mit den Transistoren TS313 und TS315 vorgeschaltet, die aus Kostengründen später offenbar entfiel. Des weiteren ist zu erkennen, daß der Monitor anfangs auch zweikanalig ausgelegt war, was beim Modell 1081 dann ebenfalls weggelassen wurde.

Beschafft man sich entsprechende Originalbauteile, kann die Zweikanaligkeit (Stereo) wiederhergestellt werden. Wie zu erfahren war, wird hierfür ein entsprechender Bausatz zum Preis von 149 Mark im Fachhandel angeboten.

So ist der Monitor dann ganz universell ausgestattet. Es fehlten nur noch Stereo- und Videotextdecoder für den Fernsehempfang (CVBS-Eingang).

(Jürgen Hirschfeld/ah)

Wenn Picasso einen Amiga gehabt hätte ...

Die Faszination, die von Grafiken eines »Super-Heimcomputers« wie des Amiga ausgehen kann, ist schon gewaltig. In diesem Kurs möchten wir Ihnen erforderliches Grundwissen über das Darstellen von Bildern mit dem Amiga vermitteln. »Mitmachen« ist angesagt!

Den Grundstein zur Computergrafik haben uns eigentlich schon die alten Griechen gelegt: Hätte da nicht ein mathematisch begabter Herr namens Euklid das Koordinatensystem erfunden, wüßte heute nämlich noch niemand, wie er seinem Computer eigentlich mitteilen könnte, daß der lilafarbene Punkt genau in der Mitte des Bildschirms erscheinen soll.

Freilich wußte Euklid noch nichts von Bits, Bytes und Bildröhren, und so wird heute aufgrund der mangelnden EDV-Bildung dieses Herrn das Koordinatensystem in der Schule noch immer falsch herum gelehrt — zumindest für Computerfreaks. Beginnt doch in der Schule dieses System links unten mit Position (0,0), obwohl der Rasterstrahl des Fernsehers das Bild von links oben nach rechts unten aufbaut (nähere Erklärung folgt).

Das kartesische Koordinatensystem, wie wir es aus der Schule kennen, zeigt uns Bild 1. Die einzelnen Einheiten des Systems haben wir in der Schule meistens in Zentimetern gemessen. Auf dem Computer teilen wir sie in »Pixels« ein; Pixel steht für »Picture Element«, also »Bildelement«. Der Einfachheit halber sagen aber sogar die Computerfachleute einfach »Punkt«, oder auf das Bild bezogen »Bildpunkt« zu so einem Element. Jeder dieser Punkte wird im Computer durch Bits dargestellt. Das bedeutet, eine 1 ist ein gesetzter Punkt und eine 0 ist ein nicht gesetzter Punkt. Aber wie schaut es mit der Farbinformation zu jedem Bildpunkt aus? Beim Amiga haben wir es mit sehr vielen Farben zu tun (maximal 4096), darum muß sich eine Farbe immer aus einer Kombination von mehreren Bits zusammensetzen, die sich eindeutig einem bestimmten Bildpunkt zuordnen lassen. Ein einzelnes Bit würde nicht ausreichen, um eine von 4096 möglichen Farben festzulegen. Beim Amiga wurde das Problem durch die sogenannten »Bitplanes« gelöst: Das sind Teile des Speichers, die alle denselben Aufbau und dieselbe Größe wie der Grafikspeicher haben. Um nun zum Beispiel den Farbwert des ersten Punktes ganz links oben zu erhalten, nimmt der Computer das diesem Punkt entsprechende Bit aus der Bitplane 1, aus der Bitplane 2 und so weiter. Diese Bits zusammengesetzt ergeben dann eine Positionsnummer innerhalb einer Tabelle, in der schließlich die gewünschte Farbinformation zu finden ist. Ob Picasso das wohl verstanden hätte?

Mit Euklid fing alles an

Der Speicher des Computers ist bekanntlich linear ab Adresse 0 aufgebaut. Die Grafik selbst (gesetzte oder nicht gesetzte Bildpunkte) haben wir durch Setzen oder Löschen von Bits in diesem Speicher definiert. Nun holt sich der Amiga jeweils ein Speicher-Word (1 Word = 16 Bit) aus dem Speicher und übergibt es seriell, also Punkt für Punkt, an den Rasterstrahl. Dieser Elektronenstrahl wandert von links oben nach rechts unten zeilenweise über den Bildschirm und zeichnet so den Speicherinhalt auf die Bildröhre.

Der Aufbau eines ganzen Bildes geschieht beim Amiga übrigens 25mal in der Sekunde (Bildwiederholfrequenz 25 Bilder/Sekunde). Der Monitor arbeitet aber doch mit 50 Hertz? So steht es zumindest hinten auf der Gehäuserückseite. Hat da jemand gemogelt? Nein, der Monitor selbst arbeitet wirklich mit 50 Hertz, aber im sogenannten »Zeilensprungverfahren«, wie jeder normale Fernseher auch: Zuerst werden vom Elektronenstrahl alle Rasterzeilen mit geraden Zeilennummern (also 2, 4, 6 und so weiter) gezeichnet; danach im zweiten Durchlauf alle »ungeraden« Rasterzeilen. Er zeichnet also zwei Halbbilder. Daraus ergibt sich eine Bildfrequenz von 25 Bildern/Sekunde.

Haben Sie schon mal Ihre Workbench auf Interlace-Modus geschaltet (im Preferences-Menü)? Das Bild besteht dann nicht mehr aus 256 sondern aus 512 Grafikzeilen, hat also die doppelte Auflösung. Aber: Das damit verbundene Bildflimmern macht ein sauberes Arbeiten fast unmöglich. Der Interlace-Modus wird softwaremäßig vom Computer erzeugt. Ein komplettes Bild besteht aus zwei Einzelbildern, die nacheinander gesendet und dargestellt werden. Natürlich halbiert sich damit wieder die Bildwiederholfrequenz: Sie sinkt auf ganze 12,5 Bilder/Sekunde. Damit das menschliche Auge jedoch ein einigermaßen stehendes Bild wahrnimmt, sind mindestens 17 Bilder/Sekunde notwendig, was auch das im Interlace-Modus entstehende Flimmern erklärt. Doch zurück zu Euklid und dem »verkehrten« Koordinatensystem.

Hätte der Erfinder der Bildröhre damals an Euklid gedacht (umgekehrt war es leider nicht möglich), und den Bildaufbau anders herum konstruiert, würden wir uns jetzt nicht immer mit der Umrechnung

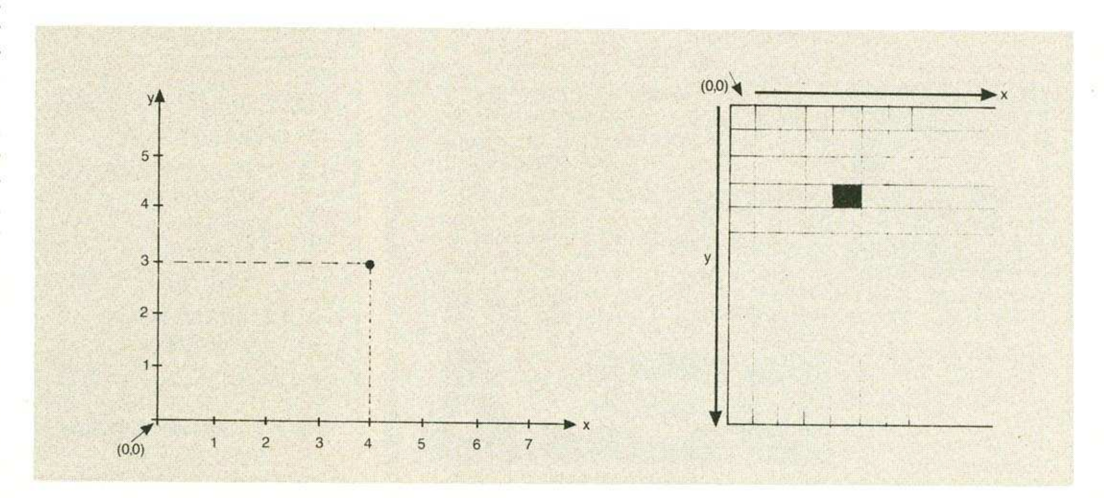


Bild 1. Ein Vergleich zwischen dem kartesischen und dem Computer-Koordinatensystem. In beiden ist der Punkt (4;3) gesetzt; links als geometrischer Punkt, rechts als Pixel.

des y-Wertes einer Grafik plagen müssen.

Bild 2 zeigt uns, wie der Bildaufbau auf dem Computer funktioniert. In Wirklichkeit läuft dieser ganze Vorgang noch wesentlich komplizierter ab, doch wir wollen hier nicht zu sehr ins Detail gehen. Aber wie Sie sehen, ist der Punkt (0,0) links oben, und nicht links unten.

Nachdem soweit die grundlegenden mathematischen und technischen Gegebenheiten geklärt sind, schreiten wir zur Praxis. Legen Sie die »Extras«-Diskette in Ihr Laufwerk und klicken das Amiga-Basic-Icon an. Vor Ihnen öffnet sich nun ein Window — und das wollen wir doch gleich als Fenster in die Grafikwelt betrachten. Geben Sie einfach mal den Basic-Befehl

```
PSET (320,100)
```

ein. Es erscheint — ein Punkt. Wenn nicht, dann schließen Sie doch erst einmal das List-Window. Da ist dann tatsächlich ein Punkt in der Mitte des Bildschirms. Sie sehen, die Grafikbefehle des Basic beziehen ihre Koordinaten nicht auf den Gesamtbildschirm, sondern auf das jeweils aktive Window. Das ist eine Eigenheit der Benutzeroberfläche (»Intuition«) des Amiga: Als Programmierer braucht man sich nicht um die Summe der Windows zu kümmern; lediglich die Angabe des mit dem jeweiligen Befehl gemeinten Windows und der Koordinaten ist notwendig. In unserem Fall ist automatisch das Ausgabe-Window das aktive Window.

Übrigens: Daß man Grafiken in einem Speicher als Pixels ablegen kann, erfaßte man erst in den 60er-Jahren unseres Jahrhunderts. Damals nannte man den Grafikspeicher noch »Frame Buffer«, wörtlich »Formenspeicher«. Damit hatte man erstmals die Möglichkeit, richtige Formen und Figuren darzustellen (davor gab es nur die Vektorgrafik, mit der man lediglich Linien darstellen konnte; der Rasterstrahl wurde einfach quer über das Bild bewegt, damit dabei eine Linie entstand).

Doch weg von den Theorien, zurück zur Praxis. Ein Punkt ist noch lange nicht faszinierend für den Betrachter. Man sollte sich aber immer bewußt sein, daß eine vollständige Grafik, so komplex sie auch erscheint, immer nur aus einzelnen Bildpunkten besteht. Eine Linie ist auch nichts weiter als eine Ansammlung von Punkten in einer bestimmten Richtung.

Es wird spannend

Ähnlich geht es mit Kreisen, Ellipsen und weitaus komplexeren geometrischen Objekten. Die Objekte bestehen immer aus einzelnen Punkten, die einer bestimmten mathematischen Formel gehorchen. Zum Glück benötigen wir als Basic-Programmierer nicht gar so viel Mathematik, denn die Autoren des Basic-Interpreters haben uns die Arbeit abgenommen und die »Algorithmen« für diese Dinge schon geschrieben. Ein Algorithmus kann eine mathematische Formel sein, aber auch eine Reihe von Bedingungen und logischen Verknüpfungen. Ein »Circle-Algorithmus« ist also die logische und mathematische Verknüpfung, die ein Programm ausführen muß, um einen Kreis zu zeichnen. Jedesmal, wenn wir mit dem Circle-Befehl einen Kreis zeichnen, beginnt der Basic-Interpreter also zu rechnen.

Was wir durch ein bißchen Mathematik und nur dem Setzen von Punkten alles anstellen können, zeigt folgendes kleines Beispiel:

Drücken Sie den rechten Mausknopf und wählen mit der Menüleiste »Windows« die Auswahl »Show List« an. Tippen Sie in dieses List-Window dann folgendes kleines Listing ein:

```
DEF FNa(x)=y-SGN(x)*SQR
    (ABS(b*x-c))
az=12000:a=333:b=-12:c=
   120
kx=250:ky=120
FOR t=1 TO az
PSET (x+kx,y+ky)
xx=FNa(x):y=a-x:x=xx
NEXT
```

Starten Sie das Programm und sehen Sie zu, was passiert. Das Muster, das hier durch das anfangs scheinbar zufällige Punktesetzen entsteht, ist durch den »Barry-Martin-Algorithmus« definiert.

Probieren Sie das gleiche mal mit den Parametern

```
az=5000:a=-150:b=0.1:c=
   -180
kx=250:ky=180
```

oder mit

```
az=10000:a=1:b=10:c=80
kx=150:ky=100
```

Ähnliche Effekte lassen sich auch durch den sogenannten »Mandelbrot«-Algorithmus definieren, der allerdings wesentlich komplizierter und aufwendiger und daher als kleines Programmbeispiel ungeeignet ist. Aber auch aufwendige 3D-Grafiken, wie zum Beispiel die »1« des ersten deutschen Fernsehprogramms ARD werden Punkt für Punkt berechnet.

Nach diesem Plädoyer für Mathematik und einzelne Punkte wenden wir uns erst einmal wieder den einfacheren Dingen des Grafiklebens zu. Wir programmieren ein Malprogramm. Der Amiga kann das in nur zwei (!) Programmzeilen:

```
malen: IF MOUSE(0)<>0
  THEN PSET (MOUSE(1),
  MOUSE(2))
  GOTO malen
```

Wenn Sie das an das vorherige Listing anhängen, können Sie die unregelmäßigen Muster, die unser Algorithmus erzeugt hat, danach noch unregelmäßiger machen.

Das Handbuch gibt Auskunft über die anderen Grafikbefehle, die das Basic zur Verfügung stellt. Die Befehle kann man unterteilen in mehrere Gruppen:

— geometrische Grundfiguren, also
- ☐ Linien mit dem LINE-Befehl
- ☐ Rechtecke ebenfalls mit LINE
- ☐ Kreise und Ellipsen mit CIRCLE
- ☐ Polygone mit den AREA-Befehlen

— Blockbefehle, wie
- ☐ SCROLL zum Bewegen von Bildteilen
- ☐ PUT und GET für Blockmanipulation

— die Animationsbefehle
- ☐ alle OBJECT-Befehle für BOBs und Sprites

Mit ein paar Befehlen, etwas Mathematik und viel Experimentierfreudigkeit kann man sogar unter Basic einige »Ahh!« und »Ohh!« hervorrufen. Dreidimensionale Objekte lassen sich beispielsweise mit ein paar LINE-Befehlen und einigen Formeln darstellen. Mehr als vier Bildschirmseiten benötigt ein Listing zur Darstellung eines 3D-»Wireframe«-Modells nicht (Wireframe= Drahtmodell, also eine dreidimensionale Figur ohne verdeckte Linien).

Vielleicht haben Sie hier gemerkt, daß das Malen nur eines einzigen Punktes schon eine Menge Wissen in sich birgt. Ausgestattet mit der notwendigen Neugier und einer gehörigen Portion an Entdeckergeist sollten Sie sich nun an die größeren Objekte wagen. Wir werden in einer der nächsten Ausgaben des Amiga-Magazins einen ausführlichen Kurs rund um das Themengebiet »Grafik« starten.

(Manfred Kohlen/tr)

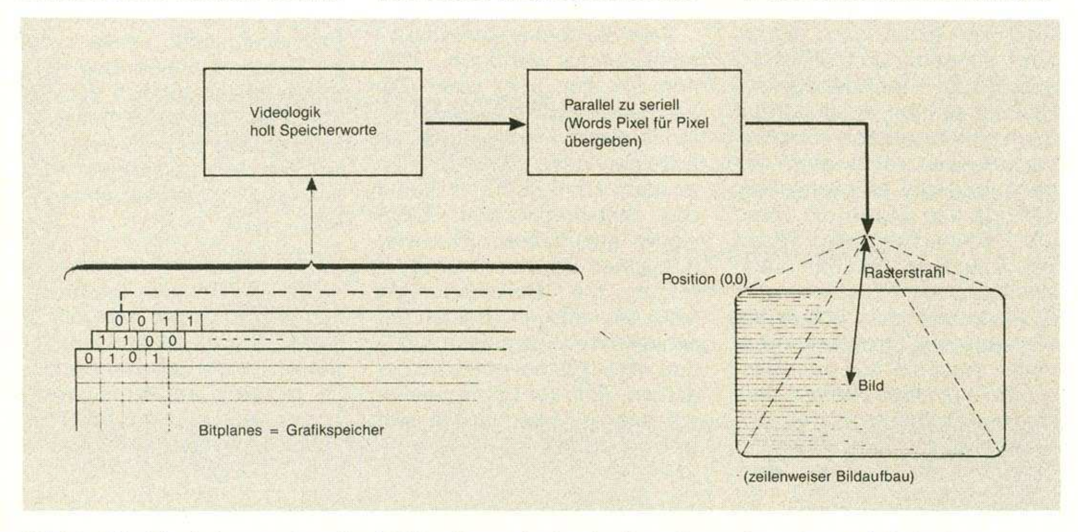


Bild 2. Die Funktionsweise des Bildaufbaus beim Amiga. Ausgehend vom Inhalt des Grafikspeichers erzeugt der Rasterstrahl das fertige Bild auf dem Monitor

CLI — Die zweite Haut des Amiga (1)

Der Amiga ist nicht nur über die Benutzeroberfläche der Workbench bedienbar. Es gibt noch eine zweite Ebene, mit der sich viel mehr anfangen läßt, als dies mit der Workbench möglich ist: das CLI. Wir starten in dieser Ausgabe einen Kurs, der Sie in den Umgang mit dem CLI einführt.

Der Amiga bietet dem Benutzer mit der Workbench eine komfortable Benutzeroberfläche. Programme können durch einfaches »anklicken« mit der Maus gestartet werden. Auch lassen sich Operationen wie beispielsweise Löschen oder Umbenennen leicht durchführen. Über die Workbench-Ebene können jedoch keine tiefergreifenden Manipulationen auf der Diskette vorgenommen werden. So läßt sich über die Workbench beispielsweise nicht feststellen, wie viele Dateien tatsächlich in welchen Schubladen (Subdirectories) auf der Diskette vorhanden sind. Oder wie wollen Sie ein Programm starten, das zwar auf der Diskette vorhanden ist, aber über kein Icon (Bildsymbol) verfügt? Doch es geht auch anders: Mit dem CLI (Command-Line-Interface, zu deutsch etwa: Kommandoübersetzer).

Doch was ist das CLI, wo ist es zu finden? Im Amiga-Handbuch stehen nicht viele Information zu diesem Programm, das sich (normalerweise unsichtbar) mit auf jeder Workbench-Diskette befindet. Bevor wir aber das CLI erklären, machen wir das Command Line Interface für den Anwender sichtbar!

Dazu fertigen Sie sich zuerst eine Arbeitskopie der Workbench-Diskette an und legen dann diese Kopie in das interne Laufwerk. Starten Sie den Amiga erneut und rufen dann das Programm »Preferences« auf. Im linken unteren Viertel des Preferences-Menüs sehen Sie ein kleines Feld, in dem »CLI ON/OFF« steht. Bei einer nicht modifizierten Workbench steht das Feld (der »Schalter«) auf »CLI OFF«. Klicken Sie nun das Wort »ON« an, speichern die Änderung mit SAVE wieder auf Diskette, bringen den Schreibschutz wieder an. Der Amiga ist anschließend neu zu starten. Klicken Sie das Symbol der Workbench-Diskette an und, nachdem das erste Fenster sichtbar ist, die Schublade mit dem Namen »System«. Durch die Änderung der Preferences findet sich nun ein neues Symbol in diesem Ordner, das den Namen »CLI« trägt. Bevor wir uns aber näher mit dem CLI befassen, soll zuvor geklärt werden, was dieses Programm eigentlich macht und wie es einzusetzen ist!

Neue Wege mit dem CLI

Von der Workbench aus kann der Benutzer nur mit Dateien umgehen, die über ein Icon verfügen. Andere Dateien können von der Workbench aus nicht beeinflußt oder aufgerufen werden. Hier greift das CLI ein: Es ermöglicht die direkte Beeinflussung aller auf einer Diskette vorhandenen Dateien. (Für alle Leser, die schon einmal mit Personal Computern gearbeitet haben: Das CLI ist in etwa vergleichbar mit MS-DOS.)

Die Befehle werden nicht (wie bei der Workbench) durch Anklicken eines Icons ausgeführt, sondern durch Kommandos, die der Anwender über die Tastatur eingibt. Versuchen wir doch gleich, anhand von praktischen Beispielen den Umgang und die Möglichkeiten, die uns das CLI in die Hand gibt, kennenzulernen. Rufen Sie dazu das Programm durch zweimaliges Anklicken des CLI-Icons in der Schublade »System« auf. Nach kurzer Ladezeit erscheint auf dem Bildschirm ein neues Fenster. Dies ist das Arbeitsfenster des CLI. Hier erfolgen alle Ein- und Ausgaben, die das Command-Line-Interface betreffen. Es empfiehlt sich, das Fenster zu vergrößern, so daß es fast den ganzen Bildschirm ausfüllt.

Anhand eines einfachen Befehls soll Ihnen nun der Unterschied zur Workbench erklärt werden. Lassen Sie dazu die Kopie der Workbench-Diskette im internen Laufwerk eingelegt und geben Sie das Kommando »dir« ein, das mit der RETURN-Taste abzuschließen ist. Nach kurzer Zeit erscheinen viele einzelne Namen in zwei Spalten auf dem Bildschirm. Dieser erste Befehl, den wir kennenlernen, hat die Aufgabe, das Directory (Inhaltsverzeichnis) der eingelegten Diskette anzuzeigen. Sie werden feststellen, daß nach Eingabe des Dir-Befehls plötzlich viel mehr Dateinamen auf dem Bildschirm erscheinen, als dies im Symbolfenster auf der Workbench der Fall wäre. Dort sehen Sie nur Dateien, die über ein Icon verfügen. Im CLI-Fenster wären das die Programme, die die Endung ».info« aufweisen (auf der Workbench-Diskette beispielsweise die Files »Preferences.info« oder »System.info«). Zu jedem dieser Icon-Files gehört auch das eigentliche Programm, welches durch das Icon aufgerufen wird (bei unserem Beispiel »Preferences« oder »System«).

Falls Sie über weitere Diskettenlaufwerke verfügen, können Sie sich über einen Zusatzparameter auch die Directories dieser Floppy-Stationen anzeigen lassen. Dazu ist hinter dem Dir-Befehl der Name des entsprechenden Laufwerks anzugeben. (Die vier möglichen Diskettenlaufwerke tragen die Bezeichnungen »df0« bis »df3«, die beiden anschließbaren Festplatten »dh0« und »dh1«.) Dabei ist nur zu beachten, daß der Laufwerksname durch ein Leerzeichen vom Befehl getrennt ist und daß am Ende ein Doppelpunkt steht. Das Directory des ersten externen Laufwerks würden Sie also mit folgendem Befehl erhalten:

```
dir df1:
```

Wenn kein externes Laufwerk angeschlossen ist, erfolgt eine entsprechende Fehlermeldung.

Da es manchmal vorkommen kann, daß mehr Dateinamen im Directory stehen, als auf dem Bildschirm angezeigt werden können, läßt sich die Ausgabe mit der Leertaste anhalten. Durch Druck auf <RETURN> oder die Backspace-Taste wird die Ausgabe fortgesetzt.

Das Directory einer Diskette besteht im allgemeinen aber nicht nur aus Dateinamen, sondern auch noch aus Subdirectories (Unterverzeichnissen). Ein Subdirectory kann man sich als Directory im Directory vorstellen. Im Hauptverzeichnis, in dem Sie sich gerade befinden, sehen Sie beispielsweise die Namen »c«, »s«, »devs« oder »Trashcan«. Hinter diesen Namen verbergen sich Subdirectories, was durch die angeschlossene Kennung »(dir)« ersichtlich ist.

Zusätze zum Dir-Befehl

Um den Inhalt eines Subdirectories anzusehen, ist ein weiterer Zusatzparameter beim Dir-Befehl nötig, nämlich der Name des Unterverzeichnisses. Möchten Sie sich etwa den Inhalt des Subdirectories »c« ansehen, ist folgendes einzugeben:

```
dir df0:c
```

Die Angabe des Laufwerknamens ist, da sowieso gerade diese Floppy-Station angewählt ist, nicht unbedingt nötig. Es ist aber empfehlenswert, immer das Laufwerk mitanzugeben, da man bei mehreren angeschlossenen Laufwerken leicht den Überblick verlieren kann.

Der Dir-Befehl bietet aber noch mehr Möglichkeiten, als

nur das komplette Directory anzuzeigen. Dazu existiert der Zusatz »opt«, der in Zusammenhang mit ihm folgenden Buchstaben den Dir-Befehl modifiziert. Durch Angabe des Buchstabens »a« hinter dem Zusatz »opt« zeigt der Amiga sämtliche Dateien auf der Diskette an, einschließlich der Namen in allen Unterverzeichnissen. Geben Sie dazu als Beispiel folgendes ein:

```
dir df0: opt a
```

Es werden sämtliche auf der Diskette enthaltene Dateinamen angezeigt. Beachten Sie bitte auch bei diesem Befehl die Schreibweise: Die einzelnen Leerzeichen dürfen nicht vergessen werden!

Gibt man anstelle des Buchstabens »a« ein »d« ein, erfolgt die Ausgabe aller Dateiverzeichnisse, also aller Namen, hinter denen die Kennung »(dir)« steht.

Ein besonderer Befehl ist der folgende:

```
dir df0: opt i
```

Tippt man diese Zeichenfolge ein, gibt der Amiga jeweils nur einen Dateinamen aus, auf den ein Fragezeichen folgt. Bei Druck auf die RETURN-Taste übergeht das CLI die entsprechende Datei und gibt den Namen der folgenden Datei aus. Stellt der angezeigte Name ein Subdirectory dar, kann durch die Eingabe von »e« in dieses gewechselt werden. <RETURN> überspringt auch hier den Dateinamen. Wurde jedoch in das Subdirectory übergewechselt und man möchte wieder in das Hauptdirectory zurück, ist »b« einzugeben. Das CLI wechselt auf die nächsthöhere Ebene zurück und macht dort weiter, wo der Einsprung in das Unterverzeichnis stattfand. Der Rest des Subdirectories wird ignoriert.

Falls hinter dem Fragezeichen die Zeichenkette »del« (»delete« = löschen) eingetippt wird, löscht dieser Befehl die betreffende Datei. Auch Dateiverzeichnisse lassen sich auf diese Weise entfernen. Dabei ist nur zu beachten, daß zuerst aus dem betreffenden Subdirectory sämtliche Dateien entfernt sein müssen!

Der Buchstabe »t« hinter »opt« schließlich erlaubt die Ausgabe des Inhalts der betreffenden Datei. Dies hat allerdings nur Sinn, wenn es sich dabei um eine Textdatei handelt. Ein Beispiel:

Sie möchten den Inhalt einer bestimmten Datei im Klartext auf dem Bildschirm ausgeben.

Viele Möglichkeiten

Nehmen wir dazu die Datei »startup-sequence«, die sich im Subdirectory »s« der Workbench-Diskette befindet. Diese Datei ist das erste File, das nach dem Einlegen der Workbenchdiskette geladen und gestartet wird (die Datei ist vergleichbar mit einer Autostart- oder Boot-Datei oder einer Batch-Datei unter MS-DOS). In dieser Datei stehen alle Befehle und Parameter, die der Amiga abarbeitet, bis nach einiger Wartezeit endlich die Workbench erscheint. Diese Startdatei besteht aus ASCII-Zeichen, ist also eine Textdatei.

Vergewissern Sie sich, daß Ihre Workbench-Kopie eingelegt ist und geben Sie folgendes ein:

```
dir df0: opt i
```

Drücken Sie nun so lange <RETURN>, bis »s (dir) ?« erscheint. Mit der Eingabe von »e« gelangen Sie nun in dieses Unterverzeichnis. Auf dem Bildschirm sollte jetzt »startup-sequence ?« sichtbar sein. Drücken Sie »t« und schließen die Eingabe mit <RETURN> ab. Blitzschnell erscheinen viele Zeilen, die wie ein Programm aussehen. Wenn Sie sich die einzelnen Zeilen ansehen, erkennen Sie einige Befehle (beispielsweise »echo«, »setmap« oder »loadwb«), die Sie ebenfalls vom CLI aus benutzen können. Auf diese Befehle soll aber erst später eingegangen werden.

Die Eingabe von »q« beendet jederzeit die Ausführung des Befehls »dir opt«. Durch die Tastenkombination <CTRL C> läßt sich ebenfalls ein Abbruch erzwingen.

Sie haben jetzt einen wichtigen Befehl kennengelernt, mit dem sich schon eine Menge anfangen läßt. Der DIR-Befehl zeigt uns aber nicht alles an. Es gibt noch weitere Informationen zu Dateien, die ein anderer Befehl viel besser anzei-

gen kann: Das Kommando »list«.

Der DIR-Befehl zeigte uns zwar alle Dateien auf der Diskette an, konnte uns jedoch keine Auskunft über deren Länge (Anzahl der Bytes) oder das Datum des letzten Updates (der letzten Speicherung) geben. (Der Amiga speichert zu jeder Datei auch die aktuelle Uhrzeit und das Datum. So läßt sich erkennen, wann die Datei das letztemal geändert wurde.) Um an diese »versteckten« Informationen zu gelangen, setzt man den Befehl »list« ein.

Versteckte Informationen

Geben Sie versuchsweise »list« gefolgt von <RETURN> ein. Zu jeder Datei des aktuellen Directories gibt das CLI jetzt zusätzliche Informationen aus. Zuerst erfährt man, wie viele Bytes in einer Datei gespeichert sind. Findet sich hier keine Angabe von Zahlen, sondern die Zeichenfolge »dir«, ist dies der Name eines Dateiverzeichnisses. Die folgenden Buchstaben »rwed« geben an, welche Operationen mit dieser Datei durchführbar sind. Dabei gilt:

»r« — die Datei kann gelesen werden
»w« — das File läßt sich überschreiben
»e« — die Datei kann als Kommando aufgerufen werden
»d« — die Daten lassen sich löschen

Im Regelfall sind alle vier Schalter auf »an«.

Die letzten beiden Zeichenblöcke geben Auskunft darüber, wann die Datei zum letztenmal gespeichert wurde (Datum und Uhrzeit). Allerdings sollten die letzten Angaben nicht zu genau genommen werden, da für eine genaue Angabe eine im Amiga eingebaute Echtzeituhr erforderlich ist, die batteriegepuffert arbeitet. Da dies aber nicht zur serienmäßigen Ausstattung des Amiga gehört, müssen diese Daten nicht unbedingt mit den Zeiten übereinstimmen, während denen die Datei gespeichert wurde.

Der list-Befehl erlaubt wie der dir-Befehl die Angabe zusätzlicher Parameter. Geben Sie versuchsweise folgendes ein:

```
list df0: p s#?
```

Das CLI zeigt Ihnen nur Dateien an, die als ersten Buchstaben ein »s« besitzen. Mit dem Parameter »p« läßt sich also eine Selektion der Dateien erzielen. Es werden nur Dateien angezeigt, die in die Maske nach dem Parameter »p« passen. Die Zeichen »#?« ersetzen den Rest des Namens, ein »?« steht für einen einzelnen Buchstaben. Einige Beispiele für zulässige Muster:

```
list df0: p sy#?
list df0: p sy??em
list df0: p #?.info
```

Das erste Beispiel würde alle Dateien anzeigen, deren Anfangsbuchstaben »sy« sind, also etwa »system« und »system.info«. Beim zweiten Befehl gibt das CLI nur Dateien auf dem Bildschirm aus, deren erste und letzten Buchstaben »sy« und »em« wären. Die beiden mittleren Zeichen unterliegen keinen Beschränkungen, könnten also Buchstaben, Zahlen oder Zeichen sein.

Der dritte Befehl zeigt alle Dateien an, deren letzte fünf Zeichen ».info« lauten (beispielsweise »system.info«, »trashcan.info« oder »preferences.info«). Diese Maskierung läßt sich auch auf den dir-Befehl anwenden. Zwei Beispiele:

```
dir df0: #?.info
dir df0: ex#?
```

Ein weiterer Parameter beim list-Befehl ist der Buchstabe »s«. Es werden nur Dateinamen ausgegeben, deren Namen den hinter »s« angegebenen Text enthalten.

```
list df0: s refer
```

würde bei eingelegter Workbench-Diskette die Dateinamen »preferences« und »preferences.info« auf dem Bildschirm ausgeben.

Sollen nur Dateien angezeigt werden, die man nach einem bestimmten Datum gespeichert hat, ist folgende Schreibweise anzuwenden:

```
list df0: since Tag-
Monat-Jahr
```

Dabei müssen die jeweiligen Daten in der Form eingegeben werden, wie sie auch bei der list-Ausgabe auf dem Bildschirm erscheinen. Ein Beispiel für eine korrekte Schreibweise:

```
list df0: since
11-jan-87
```

zeigt alle Dateien an, die nach dem 11. Januar 1987 gespeichert wurden. Das Gegenteil des Parameters »since« ist die Angabe von »upto«. In diesem Fall zeigt das CLI nur Dateien an, die bis zu einem bestimmten Datum angelegt wurden.

```
list df0: upto 11-jan-87
```

gibt alle Dateien auf dem Bildschirm aus, die vor dem 11. Januar 1987 gespeichert wurden.

Der list-Befehl kennt noch drei weitere Optionen, die man jederzeit angeben kann.

Der Zusatz »keys« zeigt zusätzlich noch die Nummer des Datei-Startblocks (File-Header) der jeweiligen Datei auf der Diskette an. Dies ist aber eine Option, die nur dann sinnvoll einzusetzen ist, wenn man über einen Diskettenmonitor und genügend Kenntnis über den Dateiaufbau auf der Diskette verfügt.

Der zweite Zusatz »nodates« unterdrückt die Ausgabe des Datums und der Uhrzeit hinter jeder Datei. Es werden also nur die Länge der Datei und die Stellung der Schalter »rwed« ausgegeben.

Der Parameter »quick« schließlich unterbindet die Ausgabe aller Zusätze. Das CLI gibt nur die Dateinamen aus, entspricht mit dem Zusatz »quick« also weitgehend dem dir-Befehl.

Der list-Befehl ist durch die Vielzahl seiner möglichen Parameter ein komplexer Befehl. Es ist also auch folgende verwirrende Kombination möglich:

```
list df0: p e#? s ref
keys since 11-mar-1986
upto 17-jan-87 nodates
quiet
```

Sie sehen, der list-Befehl erfordert etwas Training, bis man ihn beherrscht.

Die »Ebenen« der Diskette

Wie wir am Anfang dieses Artikels gelesen haben, ist der Inhalt einer Diskette meist auf verschiedene Subdirectories verteilt. Sie können inzwischen das Directory von Haupt- und Unterverzeichnissen lesen. Aber können Sie sich auch die Struktur eines Aufbaus mit Haupt- und Dateiverzeichnissen vorstellen? Deshalb zuerst eine kleine Erläuterung (betrachten Sie dazu bitte auch Bild 1):

Dieses Bild stellt einen kleinen Auschnitt der Workbench-Diskette dar. Im Bild oben sehen Sie das Hauptverzeichnis, von dem aus wir uns bis jetzt die Unterverzeichnisse mit dem dir- und list-Befehl angesehen haben. Wir hielten uns aber immer im Hauptverzeichnis auf. Zuweilen ist es aber nötig, daß wir uns in ein Subdirectory begeben, um dort zu arbeiten. Das CLI stellt für diese Aufgabe den Befehl »cd« (change directory = wechsle das Verzeichnis) zur Verfügung. Gibt man den Befehl »cd« ohne Parameter ein, so zeigt das CLI den Namen des aktuellen Dateiverzeichnisses an. Probieren wir dies gleich aus. Geben Sie bitte ein:

```
cd
```

In Ihrem CLI-Fenster müßte nun der Name der Diskette, der auch unter dem Diskettensymbol auf der Workbench zu sehen ist, erscheinen (Workbench, Workbench 1.x oder ähnlich). Wechseln Sie nun versuchsweise das aktuelle Directory. Nehmen wir an, Sie möchten in das Subdirectory »devs (dir)« überwechseln. Der Befehl dazu lautet:

```
cd devs
```

Überprüfen Sie mit

```
cd
```

ob Sie sich auch im richtigen Subdirectory befinden: In Ihrem CLI-Fenster müßte nun der Name der Diskette und, durch einen Doppelpunkt abgetrennt, der Name des Dateiverzeichnisses erscheinen, in diesem Fall »xxx:devs«. Wenn Sie Bild 1 betrachten, befinden wir uns jetzt im Kreis in der mittleren Reihe, in welchem »devs« steht. Versuchen wir den Befehl »dir«. Wir stellen fest, daß nun andere Dateien erscheinen als im Hauptdirectory. Hier finden sich drei Unterverzeichnisse mit den Namen »keymaps«, »printers« und »clipboards« (die untere Reihe in Bild 1). Mit dem Kommando

```
cd keymaps
```

würden Sie jetzt das Subdirectory »keymaps« zum aktuellen Verzeichnis erklären und befänden sich in der unteren Reihe von Bild 1. Soweit wollen wir aber nicht gehen. Wie gelangt man aber jetzt wieder nach »oben« ins Hauptdirectory? Dazu dient der Befehl

```
cd /
```

der Sie immer in das nächsthöhere (!) Directory zurückbringt. Wenn Sie sich beispielsweise in einem Subdirectory aufhielten, das vielleicht in der vierten Ebene angelegt ist, kämen Sie mit dem Befehl »cd/« nur zurück in die dritte Ebene. Um in das Hauptdirectory zu gelangen, wäre folgender Befehl nötig:

```
cd ////
```

Doch das geht auch einfacher: Mit

```
cd df0:
```

erreichen Sie immer das Hauptdirectory, egal, in welcher Tiefe Sie sich gerade aufhalten. (Sollten Sie auf einem anderen Laufwerk als »df0« arbeiten, ist natürlich immer das aktuelle Lauferk anzugeben. »cd df0:« bringt Sie nicht ins Hauptdirectory von »df1«, wenn gerade auf »df1« gearbeitet wird!)

Selbstverständlich kann auch zwischen einzelnen Directory-Zweigen gewechselt werden. Dabei muß allerdings immer der Umweg über das Hauptdirectory genommen werden. Wenn Sie sich gerade im Dateiverzeichnis »devs« aufhalten, gelangen Sie nicht mit »cd trashcan« in dieses Nebendirectory. Die korrekte Schreibweise wäre in diesem Fall

```
cd /trashcan
```

oder

```
cd df0:trashcan
```

Befindet man sich sogar eine Ebene tiefer (etwa im Subdirectory »printers«, Bild 1), wäre folgender Befehl richtig:

```
cd //trashcan
```

oder

```
cd df0:trashcan
```

Nehmen wir an, im Dateiverzeichnis »trashcan« wäre ein weiteres Subdirectory mit dem Namen »XXX« und wir wollen vom Unterverzeichnis »printers« nach »XXX« überwechseln. Dies ist etwas schwerer nachvollziehbar:

```
cd //trashcan/XXX
```

oder, etwas einfacher

```
cd df0:trashcan/XXX
```

Der cd-Befehl dient aber nicht nur zum Wechseln der Directory-Ebenen, sondern auch dem Anwählen des gewünschten Laufwerks. Nehmen wir an, das interne Laufwerk »df0« ist derzeit das aktuelle Laufwerk. Sie möchten aber mit einer Diskette arbeiten, die sich in der externen Floppy-Station befindet, also in »df1«. Das Überwechseln geschieht mit folgendem Befehl:

```
cd df1:
```

Schon arbeiten Sie mit der Diskette in »df1«. Natürlich ist es auch möglich, direkt in bestimmte Verzeichnisse zu springen:

```
cd df1:XXX
```

wobei XXX der Verzeichnisname oder eine Kombination aus Verzeichniswegen sein darf. Ein Beispiel: Das aktuelle Laufwerk ist »df1«. In »df0« befindet sich die Workbench-Diskette, von der in das gedachte Verzeichnis »XXX« im Verzeichnis »devs/printers« gewechselt werden soll. Der Aufruf sieht dann folgendermaßen aus:

```
cd df0:devs/printers/XXX
```

Schon befindet man sich im Subdirectory »XXX« im Verzeichnis »devs/printers«.

Wie bereits weiter oben erwähnt, besitzt der Amiga eine Uhr, bei der auch das Datum mit gespeichert wird. Da der Amiga normalerweise nicht über eine (nötige!) batteriegepufferte Echtzeituhr verfügt, holt sich das Betriebssystem beim Laden der Workbench das Datum der letzten Modifikation von der Diskette und verwendet diese Daten als aktuelle Zeit. Diese Uhr kann aber auch von Ihnen gestellt werden. Dazu findet sich im CLI der Befehl »date«. Ohne Parameter einzugeben erfährt man die »aktuelle« Zeit und das Datum. Sollte es nötig sein, diese Daten zu aktualisieren, muß der date-Befehl folgendermaßen eingesetzt werden:

```
date 15-apr-87
```

um das Datum zu stellen. Die Uhrzeit läßt sich genauso leicht einstellen:

```
date 01:42:13
```

Diese Schreibweise ist auch kombinierbar. Mögliche Formen:

```
date 15-apr-87 12:32:00
date 14:12 13-mar-99
```

Die Reihenfolge spielt keine Rolle. Das CLI unterscheidet Zeit und Datum anhand des Doppelpunktes und Bindestriches.

Beim nächsten Schreibzugriff auf Diskette wird die nun aktualisierte Zeit gespeichert und ist (besitzt man keine gepufferte Uhr) die Basis beim nächsten Neustart.

Es gibt aber noch einen kleinen Trick beim Setzen des Datums: Sie können das aktuelle Datum relativ leicht um einen Wochentag erhöhen oder erniedrigen. Wird beispielsweise regelmäßig jeden Tag mit einer bestimmten Diskette (idealerweise der Workbench-Kopie) gearbeitet, erhöht folgendes Kommando automatisch das aktuelle Datum um einen Tag:

```
date tomorrow
```

Schon wurde aus einem »11-06-87« das Datum »12-06-87«. Ebenso leicht kann ein Tag abgezogen werden:

```
date yesterday
```

Bindet man diesen Befehl in die »startup-sequence« ein (wie das gemacht wird, erfahren Sie in einem späteren Teil dieses Kurses), aktualisiert der Amiga bei jedem Start das Datum.

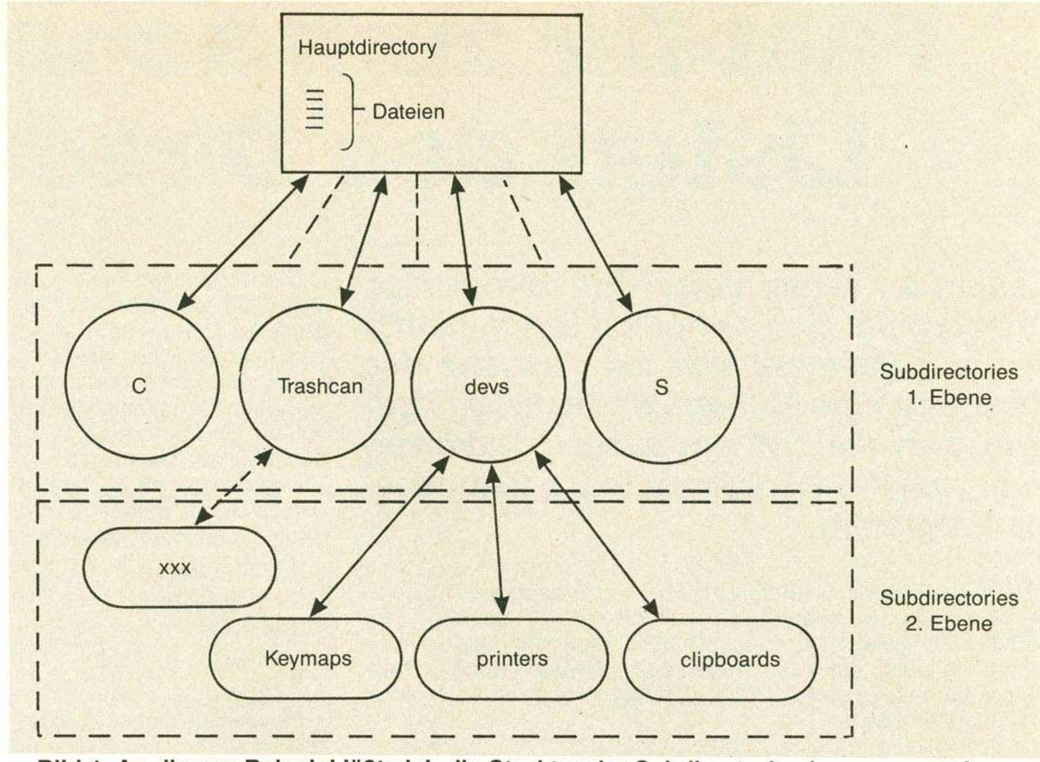


Bild 1. An diesem Beispiel läßt sich die Struktur der Subdirectories besser verstehen

Damit wollen wir den ersten Teil des CLI-Kurses abschließen. Um die Arbeit mit dem CLI zu beenden, geben Sie abschließend den Befehl »endcli« ein. Danach befinden Sie sich wieder auf der Workbench. Sie haben bereits einige Befehle kennengelernt, die Ihnen einen Eindruck von der Leistungsfähigkeit des CLI vermitteln. Sollten Sie noch Schwierigkeiten bei der Schreibweise haben, hilft die Eingabe eines durch ein Leerzeichen abgetrennten Fragezeichens hinter jedem Befehl weiter. Das CLI zeigt danach die möglichen Parameter und deren Schreibweise an.

In Teil 2 werden wir Befehle vorstellen, mit denen kopiert, gelöscht oder eine Datei umbenannt werden kann. Bis dahin, gutes Gelingen und viel Spaß mit dem Amiga. (dm)

In welchem Kursteil finde ich folgende Befehle?
Teil 1: dir, list, cd, date, endcli
Teil 2: filenote, delete, relabel, rename, type, diskcopy, info, format, install
Teil 3: makedir, ed, prompt, wait, loadwb
Teil 4: assign, echo, failat, if, say, stack

Unterstützende Literatur: Das Amiga-Handbuch, Markus Breuer, Markt & Technik-Verlag, 49 Mark

Fast-Dir zum schnellen Listen des Directory

Jeder, der schon länger mit dem Amiga arbeitet, hat sich sicherlich schon häufig darüber geärgert, daß zur Ausgabe des Directory eine Menge Zeit benötigt wird. Wie man die Zeit um einiges verkürzen kann, zeigt das folgende in C geschriebene Programm.

Es hilft nichts, die Befehle DIR und LIST zeugen nicht gerade von der Leistungsfähigkeit, die wir von unserem Computer gewohnt sind. So schnell beim Booten und so langsam beim Listen des Inhaltsverzeichnisses! Doch woher kommt dieses Verhalten eigentlich? Dazu müssen wir etwas tiefer in die Hardware einsteigen.

Wir beginnen mit einer groben Abschätzung der maximalen Geschwindigkeit der Datenübertragung:

Eine Diskette dreht sich im Amiga-Laufwerk 300mal pro Minute, benötigt also 0,2 s für eine Umdrehung.

Der zeitliche Abstand, in dem Signale auf der Magnetoberfläche gespeichert werden, beträgt überlicherweise 2 μs, folglich haben auf einer Spur 100000 dieser Signale Platz.

Aufgrund der vom Betriebssystem verwendeten MFM-Codierung werden nun zwei dieser Signale für 1 Bit benötigt. 8 Bit machen 1 Byte und so kommen wir auf eine Kapazität von 6250 Byte pro Spur. Daraus ergibt sich übrigens eine Rohkapazität der Diskette von genau einer Million Byte. Die Differenz zu der dem Benutzer zur Verfügung stehenden Datenkapazität ergibt sich aus den Aufzeichnungslücken (Gaps), Synchronisations- und Verwaltungsdaten.

Wir sprachen von 0,2 s pro Spur, es können also 31250 Byte pro s übertragen werden. Dieser Wert liegt tatsächlich zu hoch, haben wir in der Abschätzung doch vergessen, daß eine Spur überlappend gelesen werden muß, da der Schreib-/Lesekopf sich aller Wahrscheinlichkeit nach nicht gerade am Beginn einer Spur befindet. Es muß also mindestens ein Sektor mehr an Daten pro Spur gelesen werden. Weiterhin muß der Datenstrom noch decodiert und von Zeit zu Zeit der Kopf bewegt werden, was auch Zeit kostet.

Trotzdem sieht man, daß dies größenordnungsmäßig der Zeit beim Booten von »Kickstart« entspricht, denn hier können die erforderlichen Daten ohne großen Verwaltungsaufwand nahezu »roh« von der Diskette übernommen werden.

Begeben wir uns nun eine Stufe höher, und versuchen zu klären, warum nicht auch andere Daten in dieser Geschwindigkeit geladen werden können.

Das Amiga-DOS bietet, wie allgemein bekannt, ein hierarchisches Dateisystem. Es können also Verzeichnisse in Verzeichnissen und so weiter angelegt werden. Ein solches Verzeichnis läßt sich als »Baum« darstellen (siehe auch Abschnitt 1.3 »Das Dateisystem« im deutschen Handbuch zum Amiga-DOS). Trotzdem ist dadurch noch nicht festgelegt, wie die von einer Verzweigung abgehenden Äste intern verwaltet werden. Und an dieser Stelle hat man sich für ein Prinzip entschieden, das es dem Betriebssystem erlaubt, sehr schnell ein bestimmtes File zu finden und dann zu laden.

Diese als Hashing bezeichnete Methode ist jedoch zur Anzeige des Disketteninhaltsverzeichnisses weniger geeignet, da aufgrund der relativ losen Verteilung von Verwaltungsblöcken der Schreib-/Lesekopf sehr oft bewegt und manche Spuren (auf unterschiedlichen Hash-Pfaden) mehrfach gelesen werden müssen. (Für die, die es ganz genau wissen wollen, empfehlen wir an dieser Stelle die Seiten 234 bis 240 im »Amiga-DOS Manual«.)

Genau an diesem Punkt, nämlich der häufigen Bewegung des Schreib-/Lesekopfes und dem mehrfachen Lesen einer Spur greift das Programm (Listing) an: Jede gelesene Spur wird in einem internen Puffer gespeichert, so daß für den Fall, daß ein weiterer Sektor dieser Spur gelesen werden muß, auf diese Daten ohne neue Diskettenaktivität zugegriffen werden kann. Hieraus resultiert typischerweise eine rund doppelt so schnelle Ausgabe des Inhaltsverzeichnisses.

Das Programm läuft mit Workbench 1.2 und benötigt unter Lattice-C-3.10 gerade 35 s vom Source zum lauffähigen Programm. Das Batchfile sieht wie folgt aus:

```
LC1 RAM:fcat
LC2 -v RAM:fcat
BLINK LIB:c.o+RAM:fcat.o to RAM:fcat
      LIB LIB:amiga.lib+LIB:lc.lib SC SD
ND
```

Für die glücklichen Besitzer einer unter Workbench 1.2 lauffähigen »AStartup.obj« ergibt sich eine Programmlänge von unter 4 KByte, wird statt dessen (wie oben) »c.o« verwendet, so erzeugt der typische Lattice-Effekt ein Programm von gut 6 KByte, beides jedoch vertretbare Längen.

Die Syntax für »FastCatalog« lautet:

FCAT DF<drive>:[name{/name}]

mit »drive« von 0 bis 3 und »name« ein gültiger File- beziehungsweise Directory-Name unter Amiga-DOS.

Zusammen mit dem bereits genannten Abschnitt des Amiga-DOS-Manuals sowie mit Teil 2, Kapitel 7 des ROM-Kernel-Manuals »Libraries and Devices« ist das Programm leicht verständlich, trotzdem kurz zum Grobablauf:

Am Anfang wird das vom Benutzer übergebende Argument in Teilpfade zerlegt. Die Funktion »Hash()« berechnet aus einem solchen dann einen Index in der Hash-Tabelle. Die Hash-Kette wird nun so lange verfolgt, bis der gewünschte Name gefunden wurde oder kein weiterer Block mehr existiert, wobei im letzten Fall mit der Meldung »object not found« abgebrochen wird. Ansonsten wird die Blocknummer des gegebenen Pfads »Catalog()« übergeben. Durch rekursiven Aufruf hangelt »Catalog()« sich nun am Directory-Baum entlang, bis alle Einträge durchsucht sind.

Zum Schluß werden alle verwendeten »Resources« freigegeben, eine, wie ich meine, gute Idee, auf die man seine eigenen Programme einmal testen sollte!

Bei »Fast-Catalog« handelt es sich um ein kurzes Programm, das nur als Studie des Dateiverwaltungssystems unter Amiga-DOS geschrieben wurde. So bieten sich zahllose Verbesserungsmöglichkeiten an:

Pufferung der Daten: Um im Falle eines extrem gestreuten Verzeichnisses einem »Out of CHIP-Memory« zu entgehen, könnten die Daten ins FAST-Memory (sofern vorhanden) kopiert werden. Auch eine blockorientierte Pufferung, die nur Blöcke vom Typ »USERDIR« und »ROOT« speichert, wäre denkbar. So könnten dann auch bereits verwendete Blöcke sofort wieder freigegeben werden. Und sollte der Speicher wirklich trotz alledem nicht mehr ausreichen, so sollte das Programm eine ungepufferte (langsame) Bearbeitung des Verzeichnisses vornehmen.

Auch die Benutzerfreundlichkeit ist zu verbessern: Der Abbruch durch <CTRL C> ist bisher nicht möglich (kleiner Tip: die Funktion »Catalog()« ist bisher vom Typ »VOID«). Auch könnten die Einträge sortiert und, ähnlich dem LIST-Befehl, weitere Informationen zu Files und Directories ausgegeben werden.

Außerdem werden bisher weder das »Current Directory« (CD) noch »Volume Names« (ASSIGN) oder die Amiga-DOS-Wildcards akzeptiert.

Ein Programm, das all diese Funktionen besitzt, wird zwar zu lang sein, um abgedruckt zu werden, nichtsdestoweniger ein lohnendes Projekt. (Ralph Babel/ah)

```
1 /*
2 * FastCatalog.c
3 * (w) 13-Jan-1987 Guru Meditation Network
4 * Ralph Babel, Falkenweg 3, 6204 Taunusstein
5 */
6
7 /*** included files ***/
8
9 #include <exec/types.h>
10 #include <exec/io.h>
11 #include <exec/memory.h>
12 #include <exec/ports.h>
13 #include <devices/trackdisk.h>
14 #include <libraries/dos.h>
15
16 /*** external function references ***/
17
18 APTR AllocMem();
19 VOID FreeMem();
20
21 struct FileHandle *Output();
22 LONG Write();
23
24 struct MsgPort *CreatePort();
25 VOID DeletePort();
26
27 struct IOStdReq *CreateStdIO();
28 VOID DeleteStdIO();
29
30 LONG OpenDevice();
31 BYTE DoIO();
32 VOID CloseDevice();
33
34 /*** AmigaDOS constants ***/
35
36 #define NODOSBLOCK 0
37 #define MAXFILENAME 30
38
39 #define TRACKSIZE (NUMSECS * TD_SECTOR)
40 #define BLOCKSIZE (TD_SECTOR/BYTESPERLONG)
41
42 #define HASHTABLE 6
43 #define HASHEND (BLOCKSIZE - 51)
44 #define FILENAME (BLOCKSIZE - 20)
45 #define HASHCHAIN (BLOCKSIZE - 4)
46 #define SECONDARY (BLOCKSIZE - 1)
47
48 #define ST_FILE (-3)
49
50 /*** functions ***/
51
52 UBYTE capitalch(c)
53 REGISTER UBYTE c;
54  {
55  if(c >= 'a' && c <= 'z')
56   c -= 'a' - 'A';
57
58  return c;
59  }
60
61 UWORD Hash(length, s)
62 REGISTER UWORD length;
63 REGISTER UBYTE *s;
64  {
65  REGISTER UWORD hash;
66
67  for(hash = length; length--; )
68   hash = ((hash * 13 + capitalch(*s++)) & 0x7ff);
69
70  return (UWORD)(hash % 72 + HASHTABLE);
71  }
72
73 LONG *GetBlock(iosr, numtracks, map, block)
74 struct IOStdReq *iosr;
75 LONG numtracks;
76 LONG *map[], block;
77  {
78  LONG track, *ret = NULL;
79
80  if((track = block / NUMSECS) >= 0
81  && track < numtracks)
82   {
83   if(!map[track]
84   && (map[track] = (LONG *)
85      AllocMem(TRACKSIZE, MEMF_CHIP |MEMF_CLEAR)))
86    {
87    iosr->io_Command = CMD_READ;
88    iosr->io_Length = TRACKSIZE;
89    iosr->io_Data = (APTR)map[track];
90    iosr->io_Offset = track * TRACKSIZE;
91    DoIO(iosr);
92    }
93
94   if(map[track])
95    ret = map[track] + (block % NUMSECS) * BLOCKSIZE;
96   }
97
98  return ret;
99  }
100
101 VOID Catalog(nest, fh, iosr, numtracks, map, block)
102 UWORD nest;
103 struct FileHandle *fh;
104 struct IOStdReq *iosr;
105 LONG numtracks;
106 LONG *map[], block;
107  {
108  LONG *DOS;
109  UWORD i;
110
111  if(block
112  && (DOS = GetBlock(iosr, numtracks, map, block)))
113   {
114   for(i = 0; i < nest; ++i)
115    Write(fh, "  ", 2);
116   Write(fh, (UBYTE *)(DOS + FILENAME) + 1,
117         *(UBYTE *)(DOS + FILENAME));
118   if(DOS[SECONDARY] == ST_FILE)
119    Write(fh, "\n", 1);
120   else
121    {
122    Write(fh, " (dir)\n", 7);
123    for(i = HASHTABLE; i <= HASHEND; ++i)
124     Catalog(nest + 1, fh,
```

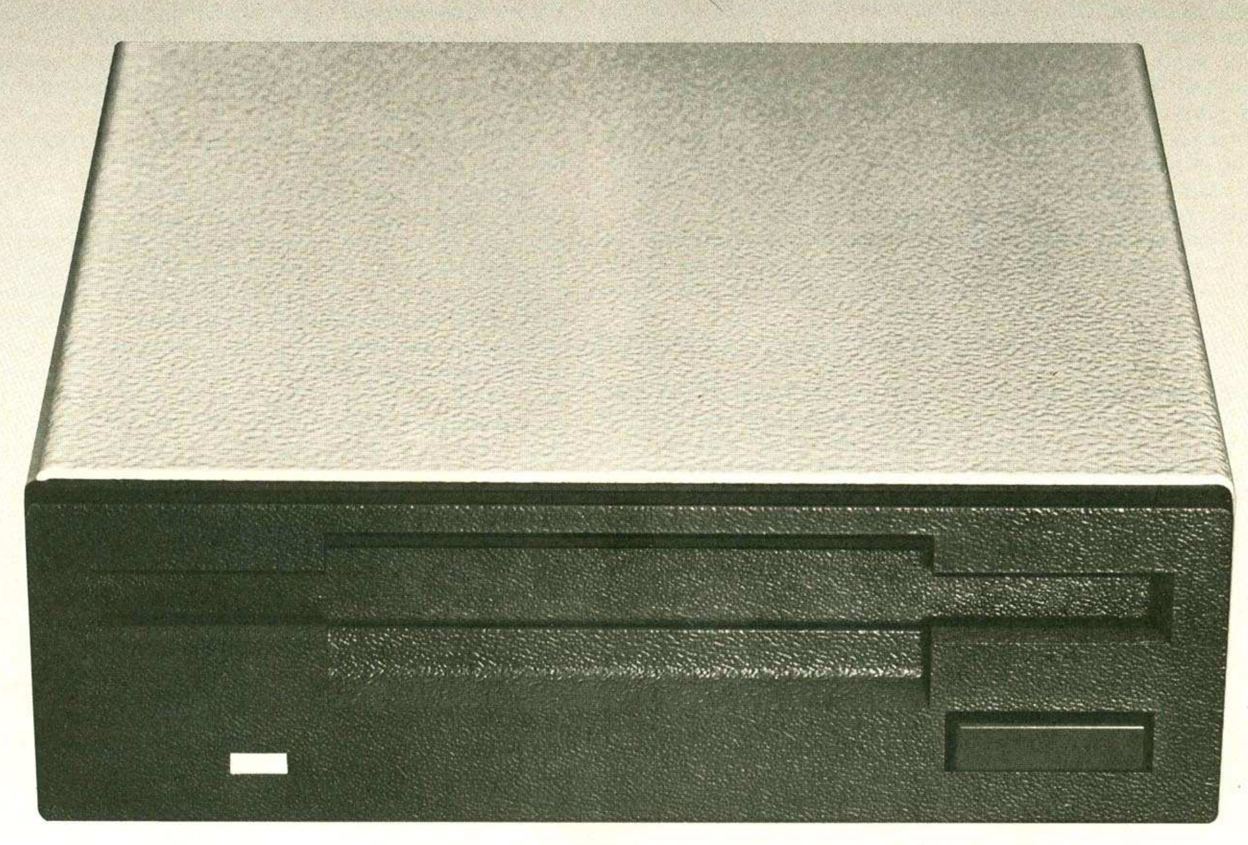
Listing. »Fast-Catalog« — ein Programm zur schnellen Ausgabe sämtlicher Directory-Einträge. Es wurde der Compiler Latice-C-3.10 verwendet.

```
125              iosr, numtracks, map, DOS[i]);
126     }
127    Catalog(nest, fh,
128           iosr, numtracks, map, DOS[HASHCHAIN]);
129   }
130 }
131
132 VOID main(argc, argv)
133 UWORD argc;
134 UBYTE *argv[];
135  {
136  UWORD drive;
137  struct FileHandle *fh;
138  struct MsgPort *mp;
139  Struct IOStdReq *iosr;
140  LONG numtracks;
141  LONG **map;
142  LONG root;
143  LONG *DOS;
144  UWORD index;
145  UWORD length;
146  UBYTE c;
147  UBYTE pathname[MAXFILENAME];
148  BOOL found;
149  UWORD i;
150
151  if(argc == 2
152  && (capitalch(argv[1][0]) == 'D'
153  && capitalch(argv[1][1]) == 'F'
154  && (drive = argv[1][2] - '0') >= 0
155  && drive < NUMUNITS)
156  && argv[1][3] == ':')
157   {
158   if(fh = Output())
159    {
160    if(mp = CreatePort(NULL, 0))
161     {
162     if(iosr = CreateStdIO(mp))
163      {
164      if(!OpenDevice(TD_NAME, drive, iosr,
165         TDF_ALLOW_NON_3_5))
166       {
167       iosr->io_Command = TD_GETNUMTRACKS;
168       DoIO(iosr);
169       numtracks = (LONG)iosr->io_Actual;
170       if(map = (LONG **)
171          AllocMem(sizeof(LONG *) * numtracks,
172                   MEMF_PUBLIC |MEMF_CLEAR))
173        {
174        DOS = GetBlock(iosr, numtracks, map,
175               root = numtracks * NUMSECS / 2);
176        index = 4;
177        while(argv[1][index])
178         {
179         if(DOS[SECONDARY] == ST_FILE)
180          {
181          Write(fh,
182         "object not of required type\n", 28);
183          goto error;
184          }
185         length = 0;
186         while(c = argv[1][index])
187          {
188          ++index;
189          if(c == '/')
190           break;
191          if(length < MAXFILENAME)
192           pathname[length++] = c;
193          }
194         root = DOS[Hash(length, pathname)];
195         do
196          {
197          if(root == NODOSBLOCK)
198           {
199           Write(fh, "object not found\n", 17);
200           goto error;
201          }
202        DOS = GetBlock(iosr, numtracks, map, root);
203        if(length == *(UBYTE *)(DOS + FILENAME))
204           {
205           found = TRUE;
206           for(i = 0; i < length; ++i)
207            if(pathname[i] !=
208            *((UBYTE *)(DOS + FILENAME) + 1 + i))
209             {
210             found = FALSE;
211             break;
212             }
213           }
214          if(!found)
215           root = DOS[HASHCHAIN];
216          }
217         while(!found);
218         }
219       Catalog(0, fh, iosr, numtracks, map, root);
220
221 error:
222       iosr->io_Command = TD_MOTOR;
223       iosr->io_Length = 0;
224       DoIO(iosr);
225       for(i = 0; i < numtracks; ++i)
226        if(map[i])
227         FreeMem(map[i], TRACKSIZE);
228        FreeMem(map, sizeof(LONG *) * numtracks);
229        }
230       CloseDevice(iosr);
231       }
232      DeleteStdIO(iosr);
233      }
234     DeletePort(mp);
235     }
236    }
237   }
238  }
```

Listing. »Fast-Catalog« — ein Programm zur schnellen Ausgabe sämtlicher Directory-Einträge (Schluß)

AMIGOS

FÜR AMIGA™



	3,5''	5 1/4''
Höhe	35 mm	45 mm
Breite	101 mm	142 mm
Tiefe	175 mm	250 mm
verwendetes Laufwerk	NEC-FD 1036 A	TEAC-FD 55 F

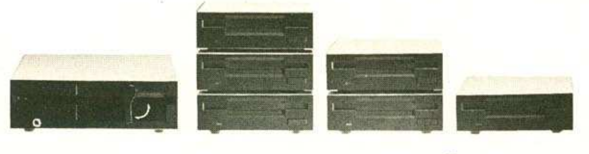
Ausbaufähig in jeder möglichen Kombination von 3 1/2'' und 5 1/4'' Laufwerken bis zu 3 externen Laufwerken auch nachträglich in wenigen Sekunden durch Steckverbindungen. Alle Laufwerke verarbeiten das 2x80 Trackformat Ihres Amigas, die 5 1/4'' Laufwerke verarbeiten zusätzlich unter MS-Dos das IBM-PC Format auf 2x40 Tracks.

Commodore Amiga, MS-Dos, NEC, TEAC und IBM sind eingetragene Warenzeichen.

Weitere Produkte wie z.B.

Harddisk, Speichererweiterung, Sampler, Midischnittstelle usw. für

Amiga 500, 1000 + 2000 auf Anfrage





Schlägel & Eisenstr. 46 • 4352 Herten 3 • Tel. 02366/55176

Coupon

Bitte schicken Sie mir Ihre komplette Hardware Information

Name ____________________

Straße ____________________

PLZ/Ort ____________________

Telefon ____________________

Font-Design

Sie möchten selbstdefinierte Zeichensätze in eigenen Basic-Programmen nutzen? Mit »Font-Design« wird Ihnen dieser Wunsch erfüllt.

Font-Design (Listing 1) wird wie jedes Basic-Programm durch Anklicken seines Icons (Symbols) geladen und gestartet. Nach kurzer Zeit blendet sich das Hauptbild ein.

Die fünf Quadrate oben in der Mitte, von denen das mittlere, hell hervorgehobene, das aktuell zu bearbeitende Zeichen ist, bilden zusammen mit den vier Shift-Feldern (», <, >, «) das Auswahlmenü. Wird eines der Felder angeklickt, so verschiebt sich der Ausschnitt der angezeigten 5 Zeichen (von insgesamt 256) nach links oder rechts. Der Unterschied zwischen Einfach- und Doppelpfeil besteht in der Geschwindigkeit, in der der Ausschnitt verschoben wird. Das große Quadrat links unten ist das Editorfeld.

Die Menüleiste enthält die zwei Menüs »Project« und »Edit«, deren Menüpunkte im folgenden beschrieben werden.

Projectmenü: In den Fällen, in denen sich der gesamte Zeichensatz verändern läßt, wird erst abgefragt, ob dieser Menüpunkt tatsächlich ausgeführt werden soll. Klickt man »no« an, so wird der Menüpunkt nicht ausgeführt.

New: löscht den gesamten Zeichensatz durch einen Neustart des Programms.

Load Editor-Set: lädt einen mit »Save Editor-Set« gespeicherten Zeichensatz. Benutzen Sie denselben Filenamen wie unter »Save Editor-Set«. Die Funktion läßt sich mit <ESC> abbrechen.

Save Editor-Set: speichert den im Computer befindlichen Zeichensatz unter dem von Ihnen angegebenen Filenamen. Ein solches File kann nicht als neuer Zeichensatz zur Darstellung von Text geladen und aktiviert werden! Der Befehl dient einzig und allein dazu, Zeichensätze extrem schnell zu speichern beziehungsweise zu laden. Zeichensätze, die zur Textdarstellung benutzt werden können, lassen sich mit dem Menüpunkt »Load Disk-Font« beziehungsweise »Save Disk-Font« laden beziehungsweise speichern. Erlaubt ist jeder gültige Filename.

Load Disk-Font: lädt einen fertigen Zeichensatz von Diskette oder kann zur Darstellung von Text geladen und aktiviert werden. Der Filename baut sich wie folgt auf: Fontname/Y-Size (zum Beispiel sapphire/14). Die Funktion läßt sich mit <ESC> abbrechen.

Save Disk-Font: erzeugt aus dem im Computer befindlichen Zeichensatz in nur wenigen Minuten einen zur Darstellung von Text ladbaren Disk-Font. Der Filename enthält im Gegensatz zum Filenamen bei »Load Disk-Font« nur die Angabe Filename (zum Beispiel »ownfont«).

End: beendet das Programm Font-Design.

Editmenü: stellt den Zeichenmodus im Editorfeld ein und enthält einige Funktionen zur Veränderung des aktuellen Zeichens.

Set, Reset und Invert dot: stellen den Zeichenmodus ein. Der Modus Set und Reset arbeiten permanent, das heißt bei gedrückter linker Maustaste, bei Invert dagegen muß die Maustaste bei jedem Punkt losgelassen werden.

Clear char: löscht das aktuelle Zeichen.

Copy char: kopiert eine Buchstabenmatrix auf eine andere, wobei zur Auswahl des Quell- und Zielbuchstabens jeweils das OK-Feld angeklickt werden muß. Ausgewählt wird im Zeichen-Menü. Die Funktion ist mit dem Auswählen des Zielbuchstabens beendet, kann aber auch vorzeitig durch Anklicken des Quit-Feldes verlassen werden.

Scrolling: dient zum Verschieben der Buchstabenmatrix innerhalb des Eingabefeldes.

Achtung: Einmal aus dem Feld geschobene Punkte sind gelöscht! Beobachten Sie die Verschiebung genau im kleinen Quadrat oben in der Mitte, oder verschieben Sie nur mit Hilfe von kurzen Mausklicks, so daß jeweils das Editor-Feld aktualisiert wird.

Load-Font: »Load-Font« (Listing 2) ist ein Unterprogramm, das dazu dient, einen beliebigen Zeichensatz von Diskette nachzuladen. Load-Font verlangt nur den Filenamen des Zeichensatzes einschließlich der Höhe als Parameter (zum Beispiel »sapphire/14«). Wird ein Leerstring als Parameter übergeben, so wird »topaz/8« aktiviert. Achten Sie darauf, daß im Hauptprogramm vor dem ersten Aufruf von Load-Font folgende drei Befehle ausgeführt werden:

```
DECLARE FUNCTION OpenDiskFont& LIBRARY
LIBRARY "diskfont.library"
LIBRARY "graphics.library"
```

Make-DFLib: Last not least gestattet das Programm »Make-DFLib« (Listing 3), die auf der Workbench gespeicherten Zeichensätze in eigene Basic-Programme einzubinden.

(Jochen Reuther/ah)

```
Achtung: Zeilennummern nicht abtippen!

11   ' vom Programm wird die "dos.library" nachgeladen.
12   ' Zu diesem Zweck muss "dos.bmap" und dieses Programm
13   ' im gleichen Directory stehen.
14
15   ' Die benutzte Diskette muss umbedingt ein
16   ' Directory ":fonts" enthalten
17
18   Mnew:
19     CLEAR ,20000&
20     CLEAR ,100000&
21     GOSUB InitAll
22     Layer$=":FonText"        ' dies ist das Directory, in dem
23     CHDIR "df0"+Layer$       ' Editorfiles abgelegt werden
24
25     ON ERROR GOTO Fehler
26
27     WHILE 1
28       m0=MENU(0) : m1=MENU(1)
29       ON m0 GOSUB ProjectMenu,EditMenu
30       m=MOUSE(0) : x=FNx(MOUSE(1)) : y=FNy(MOUSE(2))
31       IF m THEN GOSUB DoMouse
32       IF copyflag AND m>0 THEN GOSUB copiing
33     WEND
34
35
36   copiing:      ' MausKlick bei aktivierter KopierFunktion
37     IF y>98 AND y<117 THEN
38       IF x>234 AND x<276 THEN           ' Cancel angeklickt
39         copyflag=0 : GOTO ClearScreen
40       ELSEIF x>163 AND x<205 THEN      ' OK angeklickt
41         copy(copyflag)=char : copyflag=copyflag+1
42         IF copyflag=2 THEN
43           COLOR 2 : LOCATE 11,31 : PRINT char
44           COLOR 1 : LOCATE 12,20 : PRINT "copy to   :"
45         ELSEIF copyflag=3 THEN
46           IF char<>copy(1) THEN
47             FOR i=3 TO 66
48               Font(i,char)=Font(i,copy(1))
49             NEXT i
50             ShowSmall char : ShowBig char
51           END IF
52           copyflag=0 : GOTO ClearScreen
53         END IF
54       END IF
55     END IF
56   RETURN
57
58   DoMouse:    'MausKlick allgemein
59     IF x>5 AND x<6+(XSize+1)*4 AND y>49 AND y<50+(YSize+1)*4 THEN Char-
       Dot
60
61     IF y>9 AND y<28 THEN       'MausKlick in Character-Roll-Leiste
62       t!=0
63       IF x>  3 AND x< 26 AND m<0 THEN char=(char+254) MOD 256 : t!=.05
64       IF x> 26 AND x< 45 AND m<0 THEN char=(char+255) MOD 256 : t!=.2
65       IF x>285 AND x<308 AND m<0 THEN char=(char+  2) MOD 256 : t!=.05
66       IF x>266 AND x<285 AND m<0 THEN char=(char+  1) MOD 256 : t!=.2
67       IF t!>0 THEN
68         ShowSmall char
69         t!=TIMER+t! : WHILE t!>TIMER:WEND
70         m=MOUSE(0) : x=FNx(MOUSE(1)) : y=FNy(MOUSE(2))
71         IF m>=0 THEN ShowBig char
72       END IF
73     END IF
74   RETURN
75
76   CharDot:    'MausKlick im Zeichenfeld
77     x=(x-6)\4 : y=(y-50)\4
78     ON DotMode GOSUB CDset,CDreset,CDinvert
79     i=3+y*2+x\16
```

```
80    e&=2 ^(15-x MOD 16) : IF e&=32768& THEN e&=-e&
81    Font(i,char)=(Font(i,char) AND (&HFFFF XOR e&)) OR e&*c
82    IF y=BaseLine AND c=0 THEN c=2
83    LINE (7+x*4,51+y*4)-STEP(2,2),c,b
84    PUT (139,5),Font(0,char),PSET
85  RETURN
86
87  CDset: c=1 : RETURN
88  CDreset: c=0 : RETURN
89  CDinvert: c=(POINT(x*4+7,y*4+51)+(m>0)) AND 1 : RETURN
90
91  ProjectMenu:
92    copyflag=0 : GOSUB ClearScreen
93
94    IF m1<>3 AND m1<>5 THEN
95      COLOR 3
96      LOCATE 9,20 : PRINT "You will loose this"
97      LOCATE 10,20 : PRINT "font. Is that ok ?"
98      COLOR 1
99      LOCATE 12,23 : PRINT "YES      NO"
100     LINE (168,83)-STEP(40,16),2,b
101     LINE (235,83)-STEP(40,16),2,b
102     a$=" "
103     WHILE INSTR("YN",a$)=0
104       m=MOUSE(0) : x=FNx(MOUSE(1)) : y=FNy(MOUSE(2))
105       IF m>0 AND y>82 AND y<100 THEN
106         IF x>167 AND x<209 THEN a$="Y"
107         IF x>234 AND x<276 THEN a$="N"
108       END IF
109     WEND
110     GOSUB ClearScreen
111     IF a$="N" THEN RETURN
112   END IF
113
114   IF m1>1 AND m1<6 THEN
115     COLOR 1
116     LOCATE 9,20 : PRINT "type FileName"
117     LOCATE 10,20 : LINE INPUT ">";FileName$
118     IF FileName$="" THEN RETURN
119     COLOR 3
120     LOCATE 12,19 : PRINT "please wait a moment"
121   END IF
122
123   ON m1 GOSUB Mnew, MloadE, MsaveE, MloadDF, MsaveDF, Mend
124   GOTO ClearScreen
125
126 MloadE:
127   OPEN FileName$ FOR INPUT AS #1
128   IF ErrFlag=53 THEN ErrFlag=0 : RETURN
129   IF INPUT$(4,1)<>"FONT" THEN CLOSE 1 : PRINT CHR$(7) : RETURN
130   FOR i=0 TO 255
131     FOR j=0 TO 66
132       Font(j,i)=CVI(INPUT$(2,1))
133     NEXT j
134     ShowSmall i
135     IF INKEY$=CHR$(27) THEN i=255
136   NEXT i
137   CLOSE 1
138   ShowSmall char : ShowBig char
139 RETURN
140
141 MsaveE:
142   OPEN FileName$ FOR OUTPUT AS #1
143   IF ErrFlag=61 THEN ErrFlag=0 : RETURN
144   PRINT #1,"FONT";
145   FOR i=0 TO 255
146     FOR j=0 TO 66
147       PRINT #1,MKI$(Font(j,i));
148     NEXT j
149   NEXT i
150   CLOSE 1
151 RETURN
152
153 MloadDF:    'existierenden Zeichensatz laden
154   OPEN ":fonts/"+FileName$ FOR INPUT AS #1
155   IF ErrFlag=53 THEN ErrFlag=0 : RETURN
156   a$=INPUT$(LOF(1),1)
157   CLOSE 1
158   a$=MID$(a$,INSTR(a$,MKI$(&H7000)+MKI$(&H4E75))+1)
159   LOchar=ASC(MID$(a$,90))
160   HIchar=ASC(MID$(a$,91))
161   MaxY=CVI(MID$(a$,78))
162   bm=CVI(MID$(a$,96))              'modulo = LEN(Zeile)
163   bl=CVI(MID$(a$,84))              'BaseLine
164   y1=BaseLine-bl
165   fa=CVL(MID$(a$,92))              'Font MapAdresse
166   fl=CVL(MID$(a$,98))              'Font Location
167   fk=CVL(MID$(a$,106))             'Font Kern
168   FOR i=0 TO 255
169     FOR j=3 TO 66 : Font(j,i)=0 : NEXT j
170     IF i>=LOchar AND i<=HIchar THEN
171       k=(i-LOchar)*2
172       IF fk THEN x1=CVI(MID$(a$,fk+k)) ELSE x1=0
173       IF x1<0 THEN x1=0
174       bmp=CVI(MID$(a$,fl+k*2))     'BitPosition
175       bs=CVI(MID$(a$,fl+k*2+2))    'BitSize
176       FOR y=0 TO MaxY-1
177         FOR x=0 TO bs-1
178           b=-((ASC(MID$(a$,fa+y*bm+(bmp+x)\8))AND 2 ^(7-(bmp+x) MOD 8))>
179           IF b THEN
180             a=3+(y1+y)*2+(x1+x)\16
181             e&=2 ^(15-(x1+x) MOD 16) : IF e&=32768& THEN e&=-e&
182             Font(a,i)=Font(a,i) OR e&
183           END IF
184         NEXT x
185       NEXT y
186     END IF
187     ShowSmall i
188     IF INKEY$=CHR$(27) THEN i=255
189   NEXT i
190   a$=""
191   ShowSmall char : ShowBig char
192 RETURN
193
194 MsaveDF:
195   i=INSTR(FileName$,"/")
196   IF i THEN
197 MsaveDFoff:
198     LOCATE 20,20 : COLOR 3 : PRINT "bad Filename"CHR$(7)
199     t!=TIMER+3 : WHILE t!>TIMER : WEND
200     RETURN
201   END IF
202   File1$=FileName$
203   ignoreError=53
204   OPEN ":fonts/"+File1$+".font" FOR INPUT AS #1
205   ignoreError=0
206   IF ErrFlag=53 THEN
207     ErrFlag=0
208     i$=MKI$(&HF00)+MKI$(0)
209     a$=File1$+CHR$(0)
210     CHDIR ":fonts" : a&=CreateDir&(SADD(a$))
211     IF a&<>0 THEN   a&=CurrentDir&(a&)
212     CHDIR Layer$
213     OPEN ":fonts/"+File1$+".font" FOR OUTPUT AS #1
214     PRINT #1,i$;
215     CLOSE 1
216   ELSE
217     i$=INPUT$(LOF(1),1) : CLOSE 1
218   END IF
219
220   MaxX=0 : MaxY=0 : x1=XSize : x2=0 : y1=YSize : y2=0
221   LOchar=255 : HIchar=0 : bmp=0 : e=1
222   FOR i=0 TO 255
223     Font(67,i)=0 : Font(68,i)=0 : Font(69,i)=0 : Font(70,i)=0
224     FOR j=3 TO 66
225       IF Font(j,i) THEN Font(67,i)=(j-3)\2+1 : j=66
226     NEXT j
227     IF Font(67,i) THEN
228       Font(67,i)=Font(67,i)-1
229       IF LOchar>i THEN LOchar=i
230       IF HIchar<i THEN HIchar=i
231       FOR j=66 TO 3 STEP -1
232         IF Font(j,i) THEN Font(68,i)=(j-3)\2 : j=3
233       NEXT j
234       Font(69,i)=XSize
235       FOR y=Font(67,i) TO Font(68,i)
236         FOR x=0 TO Font(69,i)
237           a=3+y*2+x\16 : e&=2 ^(15-x MOD 16) : IF e&=32768& THEN e&=-e&
238           IF (Font(a,i) AND e&)<>0 AND Font(69,i)>x THEN Font(69,i)=x
              : x=XSize
239         NEXT x
240         FOR x=XSize TO Font(70,i) STEP -1
241           a=3+y*2+x\16 : e&=2 ^(15-x MOD 16) : IF e&=32768& THEN e&=-e&
242           IF (Font(a,i) AND e&)<>0 AND Font(70,i)<x THEN Font(70,i)=x
              : x=0
243         NEXT x
244       NEXT y
245       bmp=bmp+Font(70,i)-Font(69,i)+1
246       IF Font(67,i)<y1 THEN y1=Font(67,i)
247       IF Font(68,i)>y2 THEN y2=Font(68,i)
248       IF Font(69,i)<x1 THEN x1=Font(69,i)
249       IF Font(70,i)>x2 THEN x2=Font(70,i)
250     END IF
251   NEXT i
252   IF LOchar=33 THEN LOchar=32
253   bm=(bmp\16+1)*2
254   bl=BaseLine-y1
255   MaxX=x2-x1+1 : MaxY=y2-y1+1
256   IF LOchar>HIchar THEN RETURN
257
258   COLOR 1
259   LOCATE 12,19 : PRINT SPACE$(20)
260   File2$=MID$(STR$(MaxY),2)
261   LOCATE 10,21+LEN(File1$) : PRINT "/"File2$
262   IF CVI(MID$(i$,3)) THEN
263     f=0
264     FOR i=1 TO CVI(MID$(i$,3))
265       a$=MID$(i$,5+(i-1)*260,260) : a=LEFT$(a$,INSTR(a$,"/")-1)=File1$
266       IF a AND CVI(MID$(a$,257))=MaxY THEN f=5+(i-1)*260
267     NEXT i
268     IF f THEN
269       COLOR 3
```

Listing 1. »Font-Design« — für eigene Zeichensätze

```
270          LOCATE 12,20 : PRINT "Old File of this"
271          LOCATE 13,20 : PRINT "Name will be lost."
272          LOCATE 14,20 : PRINT "Is that ok ? (y/n)";
273          GOSUB JaNein
274          FOR i=12 TO 14 : LOCATE i,20 : PRINT SPACE$(20) : NEXT i : COLOR 1
275          IF a$="N" THEN RETURN
276        ELSE
277          i=CVI(MID$(i$,3))
278          i$=LEFT$(i$,2)+MKI$(i+1)+MID$(i$,5) : f=5+(i)*260
279        END IF
280      ELSE
281        i$=MKI$(&HF00)+MKI$(1) : f=5
282      END IF
283      LOCATE 12,19 : PRINT "Proportional (y,n)?";
284      GOSUB JaNein : LOCATE 12,32 : PRINT "  "a$"    "
285      LOCATE 14,19 : COLOR 3 : PRINT "please wait a moment"
286
287      Flags=&H42 : IF a$<>"N" THEN Flags=Flags OR &H20
288      a$=LEFT$(File1$+"/"+File2$+STRING$(255,0),256)
289      a$=a$+MKI$(MaxY)+CHR$(0)+CHR$(Flags)
290      i$=LEFT$(i$,f-1)+a$+MID$(i$,f+LEN(a$))
291      OPEN ":fonts/"+File1$+".font" FOR OUTPUT AS #1
292      PRINT #1,i$;
293      CLOSE 1
294
295      bm$=STRING$(bm*MaxY,0)
296      lo$="" : sp$="" : ke$="" : bmp=0
297      FOR i=LOchar TO HIchar+1
298        lo$=lo$+MKI$(bmp) : f=1
299        FOR x=Font(69,i) TO Font(70,i)
300          a1&=SADD(bm$)+bmp\8
301          e1&=2 ^(7-bmp MOD 8)
302          e2&=2 ^(15-x MOD 16) : IF e2&=32768& THEN e2&=-e2&
303          FOR y=0 TO MaxY-1
304            a2&=a1&+y*bm
305            IF Font(3+(y1+y)*2+x\16,i) AND e2& THEN
306              POKE a2&,PEEK(a2&) OR e1& : f=0
307            END IF
308          NEXT y
309          bmp=bmp+1
310        NEXT x
311        IF f THEN bmp=bmp-1
312        lo$=lo$+MKI$(bmp-CVI(RIGHT$(lo$,2)))
313        k=Font(69,i)-x1 : IF k<0 THEN k=0
314        s=MaxX
315        IF Flags AND &H20 AND i<>32 THEN s=Font(70,i)-Font(69,i)+1-(s<MaxX)
316        IF f THEN s=0
317        sp$=sp$+MKI$(s) : ke$=ke$+MKI$(k)
318      NEXT i
319
320      flen=LEN(bm$)+LEN(lo$)+LEN(sp$)
321      fa=&H6E : f1=fa+LEN(bm$) : fs=f1+LEN(lo$) : fk=fs+LEN(sp$)
322      IF Flags AND &H20 THEN
323        fk=0 : ke$=""
324      ELSE
325        fk=fs : fs=0 : sp$=""
326      END IF
327      i$=MKI$(&H7000)+MKI$(&H4E75)+MKL$(0)+MKL$(0)+MKI$(&HC00)
328      i$=i$+MKL$(&H1A)+MKI$(&HF80)+MKI$(1)+STRING$(44,0)
329      i$=i$+MKI$(&HC00)+MKL$(&H1A)+MKL$(0)+MKI$(flen)+MKI$(MaxY)
330      i$=i$+MKI$(Flags)+MKI$(MaxX)+MKI$(b1)+MKI$(1)+MKI$(0)+CHR$(LOchar)
331      i$=i$+CHR$(HIchar)+MKL$(fa)+MKI$(bm)+MKL$(f1)+MKL$(fs)+MKL$(fk)
332      i$=i$+bm$+lo$+sp$+ke$
333      WHILE LEN(i$) MOD 4 : i$=i$+CHR$(0) : WEND
334      i=LEN(i$)
335      l1$=MKL$(&H3F3)+MKL$(0)+MKL$(1)+MKL$(0)
336      l1$=l1$+MKL$(0)+MKL$(i/4)+MKL$(&H3E9)+MKL$(i/4)
337      l2$=MKL$(0)+MKL$(14)+MKL$(68)+MKL$(92)+MKL$(98)
338
339      IF Flags AND &H20 THEN i=&H66 ELSE i=&H6A
340      l2$=MKL$(&H3EC)+MKL$(5)+l2$+MKL$(i)
341      l2$=l2$+MKL$(0)+MKL$(&H3F2)
342
343      OPEN ":fonts/"+File1$+"/"+File2$ FOR OUTPUT AS #1
344      PRINT #1,l1$;i$;l2$;
345      CLOSE 1
346      KILL ":fonts/"+File1$+"/"+File2$+".info"
347    RETURN
348
349    Mend:
350      LIBRARY CLOSE
351      SCREEN CLOSE 1
352    END
353
354    ClearScreen:
355      LINE (140,64)-(311,186),0,bf
356    RETURN
357
358    JaNein:
359      a$=" "
360      WHILE INSTR("YJN",a$)=0 : a$=UCASE$(INPUT$(1)) : WEND : PRINT a$
361    RETURN
362
363    EditMenu:
364      IF m1<4 THEN
365        MENU 2,DotMode,1
366        DotMode=m1
367        MENU 2,DotMode,2
368      ELSEIF m1=5 THEN    'MenueKlick Character loeschen
369        FOR i=3 TO 66 : Font(i,char)=0 : NEXT i
370        ShowSmall char : ShowBig char
371      ELSEIF m1=6 THEN    'MenueKlick Character kopieren
372        copyflag=1
373        COLOR 1
374        LOCATE  9,20 : PRINT "use Character-Menu !"
375        LOCATE 11,20 : PRINT "copy from :"
376        LOCATE 14,23 : PRINT "OK      QUIT"
377        LINE (164,99)-STEP(40,16),2,b
378        LINE (235,99)-STEP(40,16),2,b
379      ELSEIF m1=7 THEN    'MenueKlick Bitmap scrollen
380        GOSUB InitScroll
381        GOSUB Scrolling
382        GOSUB ClearScreen
383      END IF
384    RETURN
385
386    Scrolling:
387      e=1
388      WHILE e
389        m=MOUSE(0) : x=FNx(MOUSE(1)) : y=FNy(MOUSE(2))
390        IF m THEN
391          IF x>220 AND x<243 AND y>103 AND y<126 THEN
392            e=0
393          ELSEIF x>220 AND x<243 AND y>79 AND y<102 THEN
394            FOR i=3 TO 64:Font(i,char)=Font(i+2,char):NEXT i
395            Font(65,char)=0 : Font(66,char)=0 : e=2
396          ELSEIF x>220 AND x<243 AND y>127 AND y<150 THEN
397            FOR i=66 TO 5 STEP -1:Font(i,char)=Font(i-2,char):NEXT i
398            Font(3,char)=0 : Font(4,char)=0 : e=2
399          ELSEIF x>196 AND x<219 AND y>103 AND y<126 THEN
400            FOR y=0 TO YSize
401              e=ABS((Font(4+y*2,char) AND &H8000)=&H8000)
402              t!=(Font(3+y*2,char) AND &H7FFF)*2+e : IF t!>32767 THEN t!=t!-
                 65536&
403              Font(3+y*2,char)=t!
404              t!=(Font(4+y*2,char) AND &H7FFF)*2 : IF t!>32767 THEN t!=t!-
                 65536&
405              Font(4+y*2,char)=t!
406            NEXT y : e=2
407          ELSEIF x>244 AND x<267 AND y>103 AND y<126 THEN
408            FOR y=0 TO YSize
409              e=(Font(3+y*2,char) AND 1)
410              Font(3+y*2,char)=(Font(3+y*2,char)\2 AND &H7FFF)
411              Font(4+y*2,char)=(Font(4+y*2,char)\2 AND &H7FFF) OR &H8000*e
412            NEXT y : e=2
413          END IF
414          IF e=2 THEN
415            e=1 : ShowSmall char
416            IF MOUSE(0)>=0 THEN ShowBig char
417          END IF
418        END IF
419      WEND
420    RETURN
421
422    InitScroll:
423      copyflag=0 : GOSUB ClearScreen
424      LOCATE 9,26 : COLOR 1 : PRINT "Scrolling"
425      LOCATE 15,29 : PRINT "OK"
426      LINE (221,104)-STEP(21,21),2,b   'ok
427      LINE (221,80 )-STEP(21,21),2,b   'hoch
428      LINE (221,128)-STEP(21,21),2,b   'runter
429      LINE (197,104)-STEP(21,21),2,b   'links
430      LINE (245,104)-STEP(21,21),2,b   'rechts
431      AREA (225, 97):AREA (238, 97):AREA (231, 84):AREAFILL 0
432      AREA (225,132):AREA (238,132):AREA (231,145):AREAFILL 0
433      AREA (214,108):AREA (214,121):AREA (201,114):AREAFILL 0
434      AREA (249,108):AREA (249,121):AREA (262,114):AREAFILL 0
435    RETURN
436
437    InitAll:
438      DEFINT a-z
439      DECLARE FUNCTION CreateDir& LIBRARY
440      DECLARE FUNCTION CurrentDir& LIBRARY
441      LIBRARY "dos.library"
442
443      SCREEN 1,320,200,2,1
444      WINDOW 2,SPACE$(13)+"Font-Designer",(0,1)-(311,186),0,1
445
446      FOR i=0 TO 3 : PALETTE i,0,0,0 : NEXT i
447      COLOR 3
448      LOCATE 6,9 : PRINT "please wait a moment !"
449      COLOR 1
450      FOR i!=.2 TO 1 STEP 1/15
451        PALETTE 3,i!,0,0
452        t!=TIMER+.1 : WHILE t!>TIMER : WEND
453      NEXT i!
454
455      GOSUB InitVar
456      GOSUB InitPicture
457
458      MENU 1,0,1,"Project"
459       MENU 1,1,1,"new             "
460       MENU 1,2,1,"load EditorSet"
461       MENU 1,3,1,"save EditorSet"
```

```
462     MENU 1,4,1,"load DiskFont "
463     MENU 1,5,1,"save DiskFont "
464     MENU 1,6,1,"end           "
465    MENU 2,0,1,"Edit"
466     MENU 2,1,2,"  SET dot    " : DotMode=1
467     MENU 2,2,1,"  RESET dot "
468     MENU 2,3,1,"  INVERT dot"
469     MENU 2,4,0,""
470     MENU 2,5,1,"clear char  "
471     MENU 2,6,1,"copy char   "
472     MENU 2,7,1,"scrolling   "
473    MENU 3,0,1,""
474    MENU 4,0,1,""
475
476    FOR i!=1 TO .2 STEP -1/15
477      PALETTE 3,i!,0,0
478      t!=TIMER+.1 : WHILE t!>TIMER : WEND
479    NEXT i!
480    LOCATE 6,9 : PRINT SPACE$(22)
481    ShowSmall char
482    FOR i!=.2 TO 1 STEP 1/15
483      PALETTE 1,i!,i!,i!*.8
484      PALETTE 2,i!/2,i!/2,i!/2
485      PALETTE 3,i!,0,0
486      t!=TIMER+.1 : WHILE t!>TIMER : WEND
487    NEXT i!
488
489  RETURN
490
491  InitVar:
492    XSize=31                           'Max. xsize von char
493    YSize=31                           'Max. ysize von char
494    BMSize=2+(YSize+1)*(XSize\16+1) 'Bitmap von einem char
495    DIM Font(BMSize+5,256)             'Max. 256 chars
496    FOR i=0 TO 255
497      Font(0,i)=XSize+1          'init Font fuer PUT ()
498      Font(1,i)=YSize+1
499      Font(2,i)=1
500    NEXT i
501    BaseLine=YSize*.8
502    char=32                        'aktueller char
503
504    DEF FNx(x)=x-3 : DEF FNy(y)=y-2
505  RETURN
506
507  InitPicture:
508    LOCATE 3,2 : PRINT "<< <"
509    LINE (4,10)-(25,27),2,b     'menu  <<
510    LINE (27,10)-(44,27),2,b    'menu  <
511    LOCATE 3,35 : PRINT "> >>"
512    LINE (267,10)-(284,27),2,b  'menu >
513    LINE (286,10)-(307,27),2,b  'menu >>
514
515    LINE (4,48)-(8+(XSize+1)*4,52+(YSize+1)*4),1,b
516    LINE (6,50+BaseLine*4)-STEP(XSize*4+4,4),2,bf
517
518    FOR i=-2 TO 2
519      LINE (137+i*40,3)-STEP(XSize+4,YSize+4),2+(i=0),b
520      LINE (139+i*40,5+BaseLine)-STEP(XSize,0),2
521    NEXT i
522    ShowBig 0
523  RETURN
524
525  Fehler:
526    WINDOW 2
527    ErrFlag=ERR
528    IF ERR=53 THEN
529      a$="File not found"
530    ELSEIF ERR=61 THEN
531      a$="Disk full"
532    ELSE
533      a$="Error # "+STR$(ERR)
534    END IF
535    a$=a$+CHR$(7)
536    IF ignoreError<>ErrFlag THEN
537      LOCATE 20,20 : COLOR 3 : PRINT a$
538      t!=TIMER+3 : WHILE t!>TIMER : WEND
539      IF LEFT$(a$,5)="Error" THEN STOP
540      LOCATE 20,20 : PRINT SPACE$(19)
541    END IF
542  RESUME NEXT
543
544
545  SUB ShowSmall (char) STATIC
546    SHARED Font()
547    FOR i=-2 TO 2
548      k=(char+i+256) MOD 256
549      PUT (139+i*40,5),Font(0,k),PSET
550      COLOR 2+(i=0)
551      LOCATE 6,19+i*5 : PRINT USING "###";k
552    NEXT i
553  END SUB
554
555  SUB ShowBig (char) STATIC
556    SHARED Font(),XSize,YSize,BaseLine
557    FOR y=0 TO YSize
558      FOR x1=0 TO XSize STEP 16
559        IF y=BaseLine THEN
560          c=2 : PATTERN &H7777
561        ELSE
562          c=0 : PATTERN &H8888
563        END IF
564        LINE (6+x1*4,51+y*4)-STEP(16*4,2),c,bf
565        LINE (8+x1*4,52+y*4)-STEP(16*4-2,0),2
566        IF Font(3+y*2+x1\16,char) THEN
567          FOR x=0 TO 15
568            IF Font(3+y*2+x1\16,char) AND 2^(15-x) THEN
569              PATTERN &HFFFF
570              LINE (7+(x1+x)*4,51+y*4)-STEP(2,2),1,b
571            END IF
572          NEXT x
573        END IF
574      NEXT x1
575    NEXT y
576    PATTERN &HFFFF
577  END SUB
```

Listing 1. »Font-Design« (Schluß)

```
Achtung: Zeilennummern nicht abtippen!

 1  ' die folgenden 3 Befehle muessen auf jeden Fall
 2  ' im Hauptprogramm enthalten sein !
 3
 4  DECLARE FUNCTION OpenDiskFont& LIBRARY
 5  LIBRARY "diskfont.library"
 6  LIBRARY "graphics.library"
 7
 8
 9  'dies ist ein kleines Beispielprogramm
10
11  LoadFont "robot/14"
12  CLS : PRINT "ende mit 'ESC'"
13  WHILE a$<>CHR$(27)
14    a$=INPUT$(1):PRINT a$;
15  WEND
16  LoadFont ""
17  LIBRARY CLOSE
18  END
19
20
21  SUB LoadFont (FontName$) STATIC
22
23    'ein eventuell aktiver DiskFont wird geschlossen
24    IF pFont& THEN CALL CloseFont (pFont&)
25    IF FontName$="" THEN FontName$="topaz/8"
26    i=INSTR(FontName$,"/")
27    IF i<>0 THEN     'i=0 : falscher FontName
28      a$=LEFT$(FontName$,i-1)+".font"+CHR$(0)
29      YSize=VAL(MID$(FontName$,i+1))
30      textAttr&(0)=SADD(a$)
31      textAttr&(1)=YSize*65536&
32      pFont&=OpenDiskFont&(VARPTR(textAttr&(0)))
33      'wenn ein Font dieses Namens gefunden, dann aktivieren
34      IF pFont& THEN CALL SetFont (WINDOW(8),pFont&)
35    END IF
36  END SUB
37
```

Listing 2. Mit »Load-Font« lassen sich beliebige Zeichensätze in den Amiga nachladen

```
Achtung: Zeilennummern nicht abtippen!

 1  ' Programm 'makeDFLib'
 2
 3  ' von Jochen Reuther
 4
 5  ' dieses Programm erzeugt das ausserordentlich
 6  ' nuetzliche File 'diskfont.bmap' , mit dessen
 7  ' Hilfe andere Zeichensaetze als 'topaz' zugaenglich
 8  ' werden.
 9
10  o$=""
11  FOR i=1 TO 12
12    READ a$
13    IF LEFT$(a$,1)="&" THEN
14      o$=o$+CHR$(VAL(a$))
15    ELSE
16      o$=o$+a$+CHR$(0)
17    END IF
18  NEXT i
19  OPEN "diskfont.bmap" FOR OUTPUT AS #1
20  PRINT #1,o$
21  CLOSE 1
22
23  DATA "OpenDiskFont",&hff,&he2,&h9,&h0
24  DATA "AvailFonts",&hff,&hdc,&h9,&H1,&H2,&H0
```

Listing 3. Mit »Make-DFLib« lassen sich gespeicherte Zeichensätze in eigene Basic-Programme einbinden

FiMon — ein Diskettenmonitor zum Abtippen

Um Daten auf der Diskette unmittelbar zu ändern, ist ein entsprechender Monitor erforderlich. Mit ihm lassen sich zum Beispiel die auf der Workbench gespeicherten Treiber an Ihren Drucker anpassen.

Das Programm (Listing 1) bietet die Möglichkeit, beliebige Files einzulesen, in hexadezimaler Form und ASCII-Äquivalent darzustellen und nach Veränderung auf Wunsch wieder zurückzuschreiben. Außerdem wurde eine Schnittstelle zum Amiga-DOS-Kommando-Interpreter (CLI) integriert, damit vom Basic-Programm aus zum Beispiel Sicherheitskopien von zu bearbeitenden Files erstellt werden können. Es ist ebenfalls möglich, den Teil eines Files (Block) auf dem Drucker auszugeben, der gerade bearbeitet wird. Da die Bearbeitung nicht sequentiell, sondern mit blockweisem Direktzugriff (hier: 1 Block = 256 Byte) erfolgt, reichen die File-Bearbeitungs-Befehle vom Amiga-Basic nicht aus; deshalb mußte auf die DOS.library-Funktionen zurückgegriffen werden. Bedienen läßt sich das Programm mit der Maus über Pull-Down-Menüs und mit den Funktionstasten. Um ein bestimmtes Byte in einem Block zu ändern, ist mit dem Mauszeiger einfach dieses Byte zu selektieren.

Bevor die einzelnen Befehle beschrieben werden, zuerst zu den erforderlichen Vorbereitungen, um das Programm überhaupt starten zu können. Amiga-Basic muß vom CLI aus aufgerufen werden. Vor dem ersten Start des Programms sind auf jeden Fall die erforderlichen »bmap-files« zu erzeugen. Wie das mit dem Programm »Create_b.map« (Listing 2) funktioniert, wird später noch erklärt. Nachdem dies geschehen ist, wird das Programm »FiMon« wie gewohnt geladen und gestartet. Es geschieht nun nichts Außergewöhnliches: Der Bildschirm wird gelöscht, und es erscheint ein Statusbereich in der unteren Hälfte des Bildschirms mit der Meldung »Kein File in Bearbeitung...«. Um herauszufinden, was man alles anstellen kann, sollte man nun einmal die rechte Maustaste drücken. Es erscheinen die beiden Menüleisten »OPTIONEN« und »AKTIONEN«. Schaut man sich die Menüs nun genauer an, so erhält man folgende Auswahlmöglichkeiten:

Menü OPTIONEN

OPEN (F1-Taste):
Mit OPEN (oder der Funktionstaste F1) wird ein File zur Bearbeitung geöffnet. War vorher schon ein File offen, so wird es automatisch geschlossen. In der Statuszeile erfolgt jetzt die Aufforderung zur Eingabe des Filenamens. Erlaubt ist alles, was sich an die Amiga-DOS-Konventionen hält, also auch komplette Pfadnamen (zum Beispiel df1:devs/printers/epson). Wird ein Filename angegeben, zu dem kein File existiert, erscheint eine Fehlermeldung. Das gleiche geschieht auch, wenn man anstelle eines Filenamens den Namen eines Directory angibt. Im Falle eine Fehlers wählt man die Option OPEN einfach noch einmal und versucht, diesmal etwas Vorhandenes als Filenamen einzugeben. Wurde kein Fehler gemacht, wird der erste Block des Files gelesen und auf dem Bildschirm angezeigt. Im Statusbereich wird hinter der Anzeige »BLOCKS a 256 BYTE:« die Anzahl der Blocks in diesem File angegeben. Die Zahl hinter »BLOCK:« bezieht sich immer auf die Nummer des gerade angezeigten Blocks. Schließlich wird noch das erste Byte des angezeigten Blocks invers dargestellt.

CLOSE (F2-Taste):
Mit CLOSE wird ein geöffnetes File wieder geschlossen. Wenn kein File offen ist, erfolgt eine Fehlermeldung.

CLI Command (F3-Taste):
Wird diese Option ausgewählt, so geschieht folgendes: Ein eventuell geöffnetes File wird zunächst geschlossen und ein neues CLI-Window geöffnet. Es meldet sich mit dem Prompt »3>« (der dritte CLI-Task innerhalb dieser Session). Hier können nun wie gewohnt CLI-Kommandos eingegeben werden. Will man wieder zurück ins Basic-Programm, so gibt man einfach »endcli« ein und das CLI-Window wird wieder geschlossen.

PRINT Buff (F4-Taste):
PRINT Buff gibt den Inhalt des gerade angezeigten Blocks inklusive Filename und Block-Nummer auf dem angeschlossenen Drucker aus.

QUIT (ESC):
Durch Auswahl dieses Menüpunktes oder Drücken der ESC-Taste wird das Programm augenblicklich beendet (ohne »Are You Shure«).

Menü AKTIONEN

Nächster Block (F6):
Der Block mit der nächsten Nummer wird am Bildschirm angezeigt. Sollte das Fileende schon erreicht sein, erfolgt eine Fehlermeldung.

Block zurück (F7):
Der Block mit der nächst kleineren Nummer wird angezeigt. Eine entsprechende Fehlermeldung taucht auf, wenn der Block 1 unterschritten wird.

Update Buffer (F8):
Möchte man einen veränderten Block zurückschieben, so geschieht dies mit dieser Funktion. In der Statuszeile erscheint die Meldung, wieviele Bytes zurückgeschrieben wurden.

Eingabe Blocknr. (F9):
Hier kann man direkt eingeben, welcher Block als nächstes angezeigt werden soll — ohne mühsames Hin- und Herblättern mit <F6> oder <F7>. In der Statuszeile erfolgt die Aufforderung zur Eingabe der Blocknummer. Ist diese ungültig, so erscheint eine Fehlermeldung.

Veränderung eines Byte

Bei der Anzeige eines Blocks wird immer zunächst das erste Byte invers dargestellt. Gibt man nun eine vollständige 1-Byte-Hexadezimalzahl über Tastatur ein, so wird das gerade invers dargestellte Byte entsprechend geändert. Falls möglich, wird auch das korrespondierende ASCII-Äquivalent im rechten Teil der Anzeige dargestellt. Die RETURN-Taste braucht nicht gedrückt zu werden. Jedesmal nach Eingabe zweier gültiger Hex-Ziffern (0 bis 9, A bis F) wird das gerade invers dargestellte Byte geändert.

Um nun ein anderes Byte zu ändern, ist der Mauszeiger auf das gewünschte Byte zu positionieren und die linke Maustaste zu drücken. Das vorher inverse Byte erscheint nun wieder in normaler Darstellung und das ausgewählte wird invertiert. Alle Änderungen beziehen sich jetzt auf dieses Byte. Zum Aktualisieren muß jetzt die Funktion »Update Buffer (F6)« ausgewählt werden.

So erzeugt man »bmap«-Files

Das Programm »Create_b.map« (Listing 2) wird wie gewohnt geladen und gestartet. Es meldet sich zunächst mit dem Hinweis, daß schon vorhandene »bmap-files« auf jeden Fall gesichert werden sollten und wartet dann auf einen Tastendruck. Falls mit einer anderen Taste als <F1> bestätigt wird, geschieht überhaupt nichts; das Programm springt zur Anweisung END.

Wurde mit <F1> bestätigt, so werden nun die beiden »bmaps« »exec.bmap« und »dos.bmap« erzeugt. Diese Files enthalten aber nur diejenigen LIBRARY-Funktionen, die später für den File-Monitor benötigt werden. (Ulrich Radermacher/ah)

Achtung: Zeilennummern nicht mit abtippen !

```
1 REM ***********************************************
2 REM * File Monitor fuer AMIGA 1000                *
3 REM * Kickstart V1.1                              *
4 REM * Benoetigte bmap-Files:                      *
5 REM * dos.bmap  &  exec.bmap                      *
6 REM * !!!! AmigaBasic muss vom CLI aus gestartet  *
7 REM * werden !!!!!                                *
8 REM * Autor:                                      *
9 REM * Ulrich Radermacher                          *
10 REM * Wilhelm - Nueller Str.8                    *
11 REM * 5253 LINDLAR-LINDE                         *
12 REM **********************************************
13
14
15 REM * Library Funktionen deklarieren             *
16 DECLARE FUNCTION DOpen& LIBRARY
17 DECLARE FUNCTION DRead% LIBRARY
18 DECLARE FUNCTION AllocMem& LIBRARY
19 DECLARE FUNCTION DSeek% LIBRARY
20 DECLARE FUNCTION DWrite% LIBRARY
21 DECLARE FUNCTION DLock& LIBRARY
22 DECLARE FUNCTION DExecute& LIBRARY
23 DECLARE FUNCTION DExamine& LIBRARY
24 LIBRARY "exec.library"
25 LIBRARY "dos.library"
26 DIM SHARED afeld%(15)
27
28 REM * Initialisierung der Menues                 *
29 MENU 1,0,1," O P T I O N E N "
30 MENU 1,1,1," Open        (F1) "
31 MENU 1,2,1," Close       (F2) "
32 MENU 1,3,1," CLI Command(F3) "
33 MENU 1,4,1," Print Buff (F4) "
34 MENU 1,5,1," Q U I T    (ESC) "
35 MENU 2,0,1,"  A K T I O N E N "
36 MENU 2,1,1,"  Naechster Block (F6) "
37 MENU 2,2,1,"  Block zurueck   (F7) "
38 MENU 2,3,1,"  Update Buffer   (F8) "
39 MENU 2,4,1,"  Eingabe BlockNr (F9) "
40 ON MOUSE GOSUB checkmouse
41 ON MENU GOSUB checkmenu
42
43 REM * Speicher reservieren und Variablen          *
44 REM * initialisieren                              *
45 opt&=2^0+2^16
46 bubyte&=256
47 buad&=AllocMem&(bubyte&,opt&)
48 opt&=2^1+2^16
49 inbyte&=252
50 info&=AllocMem&(inbyte&,opt&)
51 IF buad&=0 THEN ERROR 7
52 pointer=buad&:oldx=1:oldy=1
53
54 REM * Bildschirmaufbau                            *
55   CLS
56   ende=0
57   offen=0
58   notfertig=-1
59   block%=0
60   file$=""
61   MOUSE ON:MENU ON
62   LOCATE 19,10
```

Listing 1. »FiMon« — ein Diskettenmonitor für den Amiga (Fortsetzung)



```
63   LINE (3,134)-(600,151),3,bf
64   LINE (1,132)-(602,160),1,b
65   COLOR 0,3
66   LOCATE 18,2:PRINT "BLOCK  :"
67   LOCATE 18,20:PRINT "BLOCKS a 256 BYTE:"
68   LOCATE 19,2:PRINT "F-NAME :";
69   PRINT "Kein File in Bearbeitung...                     "
70   COLOR 1,0
71   LOCATE 20,2
72   PRINT"STATUS :"
73
74 REM ********************************************
75 REM * Steuerschleife                           *
76 REM ********************************************
77 loop:
78     e$=INKEY$
79     IF e$<>"" THEN taste=ASC(UCASE$(e$))
80     IF taste >47 AND taste <71 THEN
81       GOSUB change
82       taste=0
83     ELSE
84       IF taste >128 AND taste <138 THEN
85        GOSUB funktion
86        taste=0
87       END IF
88     END IF
89
90     IF taste=27 THEN notfertig=0
91     SLEEP
92     IF notfertig THEN loop
93   IF offen THEN CALL DClose(file&)
94   CALL FreeMem(buad&,bubyte&)
95   CALL FreeMem(info&,inbyte&)
96   LIBRARY CLOSE
97   END
98 REM ********************************************
99 REM *  Programm Ende                           *
100 REM ********************************************
101
102
103 REM ********************************************
104 REM * Subroutines:                             *
105 REM *                                          *
106 checkmenu:
107   num=MENU(0)
108   punkt=MENU(1)
109   IF num=1 THEN
110     IF punkt=5 THEN
111       notfertig=0
112     ELSE
113       taste=128+punkt
114     END IF
115   ELSE
116     IF num=2 THEN
117       taste=133+punkt
118     END IF
119   END IF
120
121 RETURN
122
123 REM * Ueberpruefung linke Maustaste              *
124 checkmouse:
125   IF offen THEN
126   WHILE MOUSE(0)<>0
127     xpo=MOUSE(1):ypo=MOUSE(2)
128     ych=ypo\8:xch=xpo\24
129     row=16
130     vert=actlength% \ 16
131     IF ((ych+1)*16)>actlength% THEN
132       row=actlength% MOD 16
133       vert=vert+1
134     END IF
135     IF ych<vert AND xch<row THEN
136       LOCATE oldy,oldx
137       CALL CONVERTMEM(dis$,pointer)
138       PRINT dis$
139       offset=ych*16+xch
140       pointer=buad&+offset
141       oldy=ych+1:oldx=xch*3+1
142       LOCATE oldy,oldx
143       CALL CONVERTMEM(dis$,pointer)
144       COLOR 0,1
145       PRINT dis$
146       COLOR 1,0
147     END IF
148   WEND
149   END IF
150 RETURN
151
152
153 REM ********************************************
154 REM * Aenderung eines Bytes                    *
155 change:
156 IF (taste>57  AND taste<65) OR NOT offen GOTO falsch
157   MOUSE OFF:MENU OFF
158   ncount=16
159   nibble%=15
160   GOSUB converttaste
161   nibble%=240:ncount=1
162   taste = 0
163   tas2:e$=INKEY$
164   IF e$ <>"" THEN taste=ASC(UCASE$(e$))
165   IF taste<48 OR taste>71 GOTO tas2
166   IF taste>57 AND taste<65 GOTO tas2
167   GOSUB converttaste
168   MOUSE ON:MENU ON
169   COLOR 1,0
170   ych=(offset)\16:xch=60+((offset) MOD 16):ych=ych+1
171   LOCATE ych,xch
172   IF PEEK(pointer) >32 AND PEEK(pointer) <128 THEN
173     PRINT CHR$(PEEK(pointer))
174   ELSE
175     PRINT "."
176   END IF
177
178   taste=0
179 falsch:
180 RETURN
181
182 REM * Konvertierung  ASCII-Wert in L/H-Nibble      *
183 converttaste:
184   taste=taste-48
185   IF taste >9 THEN taste = taste - 7
186   nwert%=taste*ncount
187   POKE pointer,(PEEK(pointer)AND nibble%) OR nwert%
188   CALL CONVERTMEM (dis$,pointer)
189   LOCATE oldy,oldx:COLOR 0,1:PRINT dis$
190 RETURN
191
192
193 REM ********************************************
194 REM * Aufruf der Funktion abhaengig von Funktions- *
195 REM * tasten oder Menueauswahl                 *
196 funktion:
197   MENU OFF:MOUSE OFF
198   LOCATE 20,10
199   PRINT "                                          "
200   IF taste=129 THEN
201     GOSUB oeffneFile
202   END IF
203   IF taste = 130 THEN
204     GOSUB schliesseFile
205   END IF
206   IF taste=131 THEN
207     GOSUB cli
208   END IF
209   IF taste=132 AND offen THEN
210     CALL prb(block%,file$,actlength%)
211   END IF
212   IF taste=134 AND offen THEN
```

```
213     GOSUB NextBlock
214   END IF
215   IF taste=135 AND offen THEN
216     GOSUB LastBlock
217   END IF
218   IF taste=136 AND offen THEN
219     GOSUB UpDateBuff
220   END IF
221   IF taste=137 AND offen THEN
222     GOSUB inblock
223   END IF
224
225   MENU ON:MOUSE ON
226 RETURN
227
228 REM * Eingabe des naechsten Blocks und Anzeige   *
229 inblock:
230   LOCATE 20,10
231   PRINT "Block Nr:                                    "
232   LOCATE 20,21
233   INPUT nblock%
234   corblock%=flo&\256 + 1
235   IF nblock% <1 OR nblock% >corblock% THEN
236     LOCATE 20,10
237     PRINT "Das geht aber nicht !                        "
238   ELSE
239     n&=(nblock%-block%)*256
240     oldstelle%=DSeek%(file&,n&-actlength%,0)
241     block%=nblock%
242     actlength%=DRead%(file&,buad&,256)
243     IF block%=corblock% THEN ende=-1
244     CALL DisplayBuff(buad&,actlength%)
245     LOCATE 18,10
246     COLOR 0,3:PRINT block%;"  ":COLOR 1,0
247   END IF
248 RETURN
249
250
251 oeffneFile:
252   IF offen THEN CALL DClose(file&)
253     offen=0
254     block%=0
255     LOCATE 20,10
256     LINE INPUT "File - Name :";file$
257     CALL Dexist(file$,flo&)
258     IF flo&=0 OR flo&=-1 THEN
259       LOCATE 20,10
260       PRINT "Tippfehler ?????? "
261       IF flo&=-1 THEN
262         LOCATE 20,20
263         PRINT " DIR's kann ich nicht                        "
264       END IF
265     ELSE
266
267     CALL DOpen (file$,file&)
268     offen=-1
269     actlength%=DRead%(file&,buad&,256)
270     block%=block%+1
271     IF actlength%<256 THEN ende=-1
272     CALL DisplayBuff(buad&,actlength%)
273     LOCATE 18,10
274     COLOR 0,3:PRINT block%;"  "
275     LOCATE 19,10
276     PRINT file$;"                               "
277     COLOR 1,0
278    END IF
279 RETURN
280
```

Listing 1. »FiMon« — ein Diskettenmonitor für den Amiga (Fortsetzung)

```
281 schliesseFile:
282   IF offen THEN
283     CALL DClose(file&)
284     LOCATE 19,10
285     COLOR 0,3
286     PRINT "Kein File in Bearbeitung...                    "
287     COLOR 1,0
288     offen=0
289   ELSE
290     LOCATE 20,10
291     PRINT "Kein File ist offen !?"
292   END IF
293 RETURN
294
295 REM * Aufruf von CLI-Task 3                                  *
296 cli:
297   IF offen THEN
298     CALL DClose(file&)
299     offen=0
300     LOCATE 19,10
301     COLOR 0,3
302     PRINT "Kein File in Bearbeitung...                    "
303     COLOR 1,0
304   END IF
305 wind$="newcli con:10/10/600/190/CLI"+CHR$(0)
306 s&=DExecute&(SADD(wind$),0,0)
307 IF s&=0 THEN
308   ERROR 255
309 END IF
310 RETURN
311
312 REM * Naechsten Block anzeigen                               *
313 NextBlock:
314     IF ende THEN
315       LOCATE 20,10
316       PRINT "Dateiende schon erreicht !"
317     ELSE
318       actlength%=DRead%(file&,buad&,256)
319       block%=block%+1
320       CALL DisplayBuff(buad&,actlength%)
321       LOCATE 18,10
322       COLOR 0,3:PRINT block%;"  ":COLOR 1,0
323       IF actlength%<256 THEN ende=-1
324     END IF
325 RETURN
326
327 REM * Vorherigen Block anzeigen                              *
328 LastBlock:
329     IF block% >1 THEN
330       oldstelle%=DSeek%(file&,-256-actlength%,0)
331       block%=block%-1
332       actlength%=DRead%(file&,buad&,256)
333       ende=0
334       CALL DisplayBuff(buad&,actlength%)
335       LOCATE 18,10
336       COLOR 0,3:PRINT block%;"  ":COLOR 1,0
337     ELSE
338       LOCATE 20,10
339       PRINT "Block 1 schon erreicht !"
340     END IF
341 RETURN
342
343 REM * Block mit Veraenderungen zurueckschreiben             *
344 UpDateBuff:
345       oldstelle%=DSeek%(file&,-actlength%,0)
346       retlength%=DWrite%(file&,buad&,actlength%)
347       IF retlength% <0 THEN
348         LOCATE 20,10
349         PRINT "Fehler beim Schreiben !"
350       ELSE
351         LOCATE 20,10
352         PRINT "Es wurden ";retlength%;
353         PRINT " Bytes geschrieben !"
354       END IF
355 RETURN
356 REM *                                                          *
357 REM * Ende Subroutines                                         *
358 REM ***********************************************************
359
360
361 REM ***********************************************************
362 REM * Unterprogramme:                                          *
363 REM *                                                          *
364 SUB DOpen(execute$,f&) STATIC
365 file$=execute$+CHR$(0)
366 f&=DOpen&(SADD(file$),1005)
367 IF f&=0 THEN
368   ERROR 255
369 END IF
370
371 END SUB
372
373 REM * Ueberpruefung ob File vorhanden                          *
374 SUB Dexist(execute$,fl&) STATIC
375 SHARED info&
376 fil$=execute$+CHR$(0)
377 access%=-2
378 lock&=DLock&(SADD(fil$),access%)
379 IF lock&=0 THEN
380   fl&=0
381 ELSE
382   suc&=DExamine&(lock&,info&)
383   fl&=PEEKL(info&+4)
384   IF fl& >0 THEN
385     fl&=-1
386   ELSE
387     fl&=PEEKL(info&+124)
388     LOCATE 18,40
389     COLOR 0,3
390     PRINT " ";fl&\256 + 1;"  "
391     COLOR 1,0
392   END IF
393
394 END IF
395 CALL DUnLock(lock&)
396 END SUB
397
398 REM ***  Anzeige des aktuellen Buffer-Inhalts                  *
399 SUB DisplayBuff(baddress&,length%) STATIC
400 SHARED oldx,oldy,pointer
401 oldx=1:oldy=1:pointer=baddress&
402 i=0:sasc=60:z=1:n=0
403 LINE (0,0)-(619,130),0,bf
404 LOCATE 1,1
405 MOUSE OFF
406 WHILE i<length%
407   wert=PEEK(baddress&+i)
408   afeld%(n)=wert
409   n=n+1
410   i=i+1
411   IF wert<16 THEN
412     dis$="0"+HEX$(wert)
413    ELSE
414     dis$=HEX$(wert)
415   END IF
416
417   PRINT dis$" ";
418   IF i MOD 16 = 0 THEN
419    LOCATE z,60
420    FOR n=0 TO 15
421      wert=afeld%(n)
422      IF (wert<32) AND (wert>128) THEN
423        PRINT CHR$(wert);
424      ELSE
425        PRINT ".";
426      END IF
427    NEXT n
428    PRINT
429    z=z+1
430    n=0
```

```
431   END IF
432 WEND
433 IF n>0 THEN
434   LOCATE z,60
435   FOR i=0 TO n-1
436      wert=afeld%(i)
437      IF (wert>32) AND (wert<128) THEN
438        PRINT CHR$(wert);
439      ELSE
440        PRINT ".";
441      END IF
442   NEXT i
443 END IF
444 pointer=baddress&
445 CALL CONVERTMEM(dis$,pointer)
446 COLOR 0,1:LOCATE 1,1:PRINT dis$:COLOR 1,0
447 MOUSE ON
448 END SUB
449
450 REM * Umwandlung Peek-Wert in Hexadezimal String  *
451 SUB CONVERTMEM (show$,zeiger) STATIC
452    wert=PEEK(zeiger)
453    IF wert<16 THEN
454      show$="0"+HEX$(wert)
455    ELSE
456      show$=HEX$(wert)
457    END IF
458 END SUB
459
460 REM * Ausgabe Buffer -->Drucker                    *
461 SUB prb (bln%,fi$,actl%) STATIC
462 SHARED buad&
463   OPEN "prt:" FOR OUTPUT AS #1
464   PRINT #1,"FILE :";fi$
465   PRINT #1,CHR$(13)
466   PRINT #1,"Hex Dump  von  Block Nr: ";bln%
467   PRINT #1,CHR$(13)
468   n=0
469   FOR i%=1 TO actl%
470   wert=PEEK(buad&+i%-1)
471   afeld%(n)=wert
472   n=n+1
473   IF wert<16 THEN
474     dis$="0"+HEX$(wert)
475    ELSE
476     dis$=HEX$(wert)
477   END IF
478
479   PRINT #1,dis$" ";
480   IF i% MOD 16 = 0 THEN
481    PRINT #1,"     ";
482    FOR n=0 TO 15
483      wert=afeld%(n)
484      IF (wert>32) AND (wert<128) THEN
485        PRINT #1,CHR$(wert);
486      ELSE
487        PRINT #1,".";
488      END IF
489    NEXT n
490    n=0
491    PRINT #1,CHR$(13)
492   END IF
493   NEXT i%
494   CLOSE #1
495 END SUB
496
497 REM * Ende Unterprogramme und Listing                *
498 REM ************************************************
```

Listing 1. »FiMon« — ein Diskettenmonitor für den Amiga (Schluß)

```
 1 REM *******************************************
 2 REM * .bmap Generator                          *
 3 REM * Autor:                                   *
 4 REM * Ulrich Radermacher                       *
 5 REM * Wilhelm-Mueller Str. 8                   *
 6 REM * 5253 Lindlar-Linde                       *
 7 REM *******************************************
 8 PRINT " bmap Generator erzeugt die fuer den"
 9 PRINT " File Monitor benoetigten bmap FILES"
10 PRINT
11 PRINT "  !!!!  W I C H T I G !!!!!"
12 PRINT
13 PRINT " Sollten sich exec.bmap und dos.bmap"
14 PRINT " bereits auf Diskette befinden, so"
15 PRINT " muessen diese unbedingt vorher ge-"
16 PRINT " sichert werden !"
17 PRINT " (rename oder copy TO andere_Disk)"
18 PRINT " Ansonsten werden sie ueberschrieben !"
19 PRINT
20 PRINT " Alles klar ? "
21 PRINT " Wenn ja, druecke F1 "
22 warte:e$=INKEY$
23 IF e$="" THEN warte
24 IF e$ <>CHR$(129) THEN PRINT "Nix passiert!":GOTO nix
25 PRINT "xxxx.bmap Generator"
26 DATA "dos.bmap"
27 DATA DClose,2,36
28 DATA DExamine,2,3,102
29 DATA DLock,2,3,84
30 DATA DOpen,2,3,30
31 DATA DRead,2,3,4,42
32 DATA DSeek,2,3,4,66
33 DATA DUnLock,2,90
34 DATA DWrite,2,3,4,48
35 DATA DExecute,2,3,4,222
36 DATA "exec.bmap"
37 DATA AllocMem,1,2,198
38 DATA FreeMem,10,1,210
39 DATA ende
40
41
42 READ nam$
43 WHILE nam$ <>"ende"
44   OPEN nam$ FOR OUTPUT AS 1
45   READ nam$
46   Gesamt$=""
47   WHILE nam$ <>"exec.bmap" AND nam$ <>"ende"
48     regist$=""
49     READ wert
50     WHILE wert <20
51       regist$=regist$+CHR$(wert)
52       READ wert
53     WEND
54     wert=65536&-wert
55     HByte=INT(wert/256)
56     LByte=wert-(256*HByte)
57     HByte$=CHR$(HByte):LByte$=CHR$(LByte)
58     Funkt$=nam$+CHR$(0)+HByte$+LByte$+regist$+CHR$(0)
59     Gesamt$=Gesamt$+Funkt$
60     READ nam$
61   WEND
62   PRINT #1,Gesamt$
63   CLOSE 1
64 WEND
65 nix:
66 END
```

Listing 2. Da die herkömmliche »dos.bmap« nicht so funktioniert wie sie sollte, läßt sich mit »Create_b.map« eine neue funktionierende Datei anlegen

Ein etwas anderer Line-Befehl

Sie möchten vom Amiga-Basic aus rotierende Drahtmodelle auf den Bildschirm zaubern? An einem kleinen Beispiel zeigen wir Ihnen, wie man so etwas programmiert und wie es funktioniert.

Das Betriebssystem des Amiga enthält zwei Routinen zum Zeichnen von Linien, die »Draw«-Routine und die »PolyDraw«-Routine. Für die Programmierung in Basic ist die Draw-Funktion nicht interessant, da sie für die Basic-Anweisung Line verwendet wird. Sie zieht lediglich eine Linie von der aktuellen Zeichenposition zu einer übergebenen x/y-Koordinate.

Die »PolyDraw«-Funktion eröffnet dagegen für den Basic-Programmierer Möglichkeiten, die sich durch keine Amiga-Basic-Anweisung ersetzen lassen. »PolyDraw« verbindet eine Reihe von x/y-Positionen, die in einem Integer-Datenfeld übergeben werden. Die Adresse des Datenfeldes läßt sich mit der VARPTR-Funktion übergeben. Außerdem muß der »PolyDraw«-Routine die Anzahl der zu zeichnenden Linien und die Adresse des »RastPort«-Datensatzes des Ausgabefensters mitgeteilt werden. Die »RastPort«-Adresse muß man mit der Funktion WINDOW(8) ermitteln. Ein Aufruf der »PolyDraw«-Funktion sieht also folgendermaßen aus:

```
call PolyDraw ( RastPort, count, array )
RastPort = RastPort-Adresse des Ausgabefensters
count    = Anzahl der Linien
array    = Zeiger auf die erste X-Position im
           Datenfeld
```

Dabei ist zu beachten, daß die Adresse von »array« im Gegensatz von »RastPort« und »count« nicht als Variable übergeben werden darf. Aufgrund der dynamischen Speicherverwaltung verschiebt sich das Datenfeld bei jeder neuen Definition. Wenn »PolyDraw« zum Beispiel eine Grafik zeichnen soll, die aus zwanzig Linien besteht und deren x/y-Positionen im Feld a%(0) bis a%(39) abgelegt sind, so kann man schreiben:

```
RastPort& = WINDOW(8)
count%    = 20
CALL PolyDraw ( RastPort&, count%. VARPTR(a%(0))
```

oder

```
CALL PolyDraw (WINDOW(8). 20. VARPTR(a%(0))
```

Da »PolyDraw« in der Bitmap zeichnet, kann die Zeichnung größer sein als der momentan angezeigte Bildausschnitt. Sie ist damit nicht auf 640 x 200 Punkte begrenzt. Die Werte für die x/y-Positionen können Werte zwischen 0 und 1024 annehmen. Wenn sie außerhalb dieses Bereichs liegen, wird automatisch ein Clipping der Linien an den Kanten der Bitmap durchgeführt. Bei Verwendung von berechneten Koordinaten, braucht das aufrufende Programm also nicht zu prüfen, ob die Zeichnung überhaupt in die Bitmap hineinpaßt. Da »PolyDraw« die Linie jeweils zur nächsten x/y-Position weiterzieht, kann mit einem Funktionsaufruf nur eine zusammenhängende Linienfolge gezeichnet werden. Wenn man an einer Position weiterzeichnen möchte, muß »PolyDraw« erneut aufgerufen werden. Die Startposition der Linienfolge wird mit der »Move«-Funktion des Betriebssystems festgelegt. Der Aufruf sieht folgendermaßen aus:

```
CALL Move ( RastPort.XPos.YPos)
```

Die »PolyDraw«-Routine nutzt die hohe Geschwindigkeit des Systems voll aus. Komplexe Zeichnungen lassen sich mit wenigen System-Aufrufen realisieren. Die Zeichengeschwindigkeit ist dadurch um ein Vielfaches höher als das Zeichnen mit der Line-Anweisung des Basic-Interpreters.

Da nur die Endpunkte der Linien gespeichert werden, können umfangreiche Zeichnungen in nur wenigen Speicherzellen abgelegt werden. Durch Umrechnen dieser Koordinaten kann man zum Beispiel Objekte aus verschiedenen Blickrichtungen abbilden. Sie lassen sich auch vergrößern, verkleinern oder rotieren.

Da der Speicherbedarf dieser »Drahtmodelle« außerordentlich gering ist, kann man ganze Bildfolgen berechnen und in einem Datenfeld speichern. Wenn diese Bildfolgen nacheinander gezeichnet werden, lassen sich Bewegungsabläufe weitgehend fließend auf dem Bildschirm darstellen.

Die Koordinaten werden im Datenfeld im 2-Byte-Integer-Format abgelegt. Das erste Wort enthält die erste x-Position, das zweite Wort die erste y-Position, das dritte die zweite x-Position und so weiter. Im Basic sieht das dann folgendermaßen aus: a%(0)=x1,a%(1)=y1,a%(3)=y2.... Bildfolgen können in einem Feld hintereinander abgelegt werden. Wenn das erste Bild zum Beispiel von a%(0) bis a%(49) gespeichert ist, kann das zweite Bild von a%(50) bis a%(99) liegen. Beim »PolyDraw«-Aufruf wird dann die Adresse des Datensatzes des ersten Bildes mit VARPRT (a%(0)), der Datensatz des zweiten mit VARPTR(a%(50)) übergeben.

Dreidimensionale Drahtmodelle

»PolyDraw« und »Move« sind Bestandteil der Grafik-Bibliothek des Amiga. Um einen Zugriff auf die Routinen zu ermöglichen, muß die Bibliothek mit der LIBRARY-Anweisung des Basic geöffnet werden. Dazu muß eine Grafik.bmap-Datei auf der Diskette existieren und im aktuellen Directory oder im Libs-Directory liegen. Sie kann mit dem ConvertFD-Programm von der Extras-Diskette erzeugt oder einfach aus der Basic-Demo-Directory kopiert werden — dort existiert sie nämlich schon.

Programmbeschreibung zum »PolyDraw«-Demo (Listing)

Zuerst werden die Felder für die beiden Bildfolgen dimensioniert. Die 40 Einzelbilder benötigen nur zirka 8 KByte. Danach wird die Grafik-Bibliothek geöffnet. Beachten Sie, daß in der LIBRARY-Anweisung als Dateiname »graphics.library« angegeben wird, während die Datei auf der Diskette »graphic.bmap« heißt.

Die erste FOR/NEXT-Schleife liest die Koordinaten des Bildes in die Variablen a%(0) bis a%(47). In der zweiten Schleife werden die x/y-Koordinaten umgerechnet, um das Bild zu verkleinern. Die Einzelbilder stehen in den Variablen von a%(50) bis a%(1997). Die dritte Schleife berechnet eine Bildfolge, die das Bild um die x-Achse rotieren läßt. Die Daten der Bildfolge befinden sich in den Variablen von b%(0) bis b%(1797).

Die Programmausführung erfolgt ab Label »f5« und gliedert sich in drei Teile. Im ersten Teil wird das Bild vergrößert und von oben links zur Mitte bewegt. Im zweiten Teil rotiert es um die x-Achse. Im dritten Teil wird es wieder verkleinert und nach oben links geschoben. Vor dem Zeichen des nächsten Bildes wird das vorherige gelöscht, indem es in der Hintergrundfarbe überschrieben wird. Dieses hat sich als schneller und flackerfreier erwiesen, als ein Löschen mit der CLR-Funktion.

Ein Tip für Assembler-Freaks: Assemblerprogrammierer übergeben die »RastPort«-Adresse in A0, count in D0 und array in A1 an »PolyDraw«. An Move wird die »RastPort«-Adresse in A1 übergeben, die x-Position in D0 und die y-Position in D1.

(Gerhard Ross/ah)

```
Achtung: Zeilennummern nicht mit abtippen!

 1     ' PolyDraw Demo
 2     ' ------------------------------
 3     ' Gerhard Ross  * 11.Jan.1987
 4     '------------------------------
 5     'Dieses Programm demostriert den Einsatz der
 6     'PolyDraw - Routine des Betriebssystems
 7
 8      DEFLNG a-z
 9      DIM a%(2052),b%(1852)
10      LIBRARY "graphics.library"
11      RastPort=WINDOW(8)
12      pi=3.141593
13
14      'Koordinatnn einlesen
15        FOR i=0 TO 47 : READ a%(i) : NEXT
16
17      'Zoomkoordinaten berechnen
18        FOR j=50 TO 2000 STEP 50
19          FOR i=0 TO 47
20            a%(j+i)=INT(a%(j+i-50)*.97)
21          NEXT
22        NEXT
23
24      'Rotationskoordinaten berechnen
25        FOR i=0 TO 1800 STEP 50
26          FOR j=0 TO 50 STEP 2
27            b%(i+j+1)=(a%(j+1)-100)*SIN((i/5+90)*pi/180)+100
28            b%(i+j)=a%(j)
29          NEXT
30        NEXT
31
32 f5:  CLS
33        FOR i=2000 TO 0 STEP -50
34          COLOR 0
35          CALL Move (RastPort,a%(i+50),a%(i+51))
36          CALL PolyDraw(RastPort,23,VARPTR(a%(i+52)))
37          COLOR 1
38          CALL Move (RastPort,a%(i),a%(i+1))
39          CALL PolyDraw(RastPort,23,VARPTR(a%(i+2)))
40        NEXT
41
42        FOR j=1 TO 2
43          CLS
44          FOR i=1800 TO 0 STEP -50
45            COLOR 0
46            CALL Move(RastPort,b%(i+50),b%(i+51))
47            CALL PolyDraw(RastPort,23,VARPTR(b%(i+52)))
48            COLOR 1
49            CALL Move(RastPort,b%(i),b%(i+1))
50            CALL PolyDraw(RastPort,23,VARPTR(b%(i+2)))
51          NEXT
52        NEXT
53
54        FOR i=50 TO 2000 STEP 50
55          COLOR 0
56          CALL Move (RastPort,a%(i-50),a%(i-50+1))
57          CALL PolyDraw(RastPort,23,VARPTR(a%(i-50+2)))
58          COLOR 1
59          CALL Move (RastPort,a%(i),a%(i+1))
60          CALL PolyDraw(RastPort,23,VARPTR(a%(i+2)))
61        NEXT
62
63      GOTO f5
64
65     DATA 202,112,240,150,260,150,360,50,340,50,265,125
66     DATA 250,110,230,110,270,150,290,150,390,50,370,50
67     DATA 270,150,290,150,265,125,240,150,260,150,220,110
68     DATA 200,110,200,160,400,160,400,40,200,40,200,110
69
70
```

Listing. Demoprogramm zur »PolyDraw«-Anweisung

Netzgrafik mit ABasic

Die Darstellung von dreidimensionalen Grafiken ist immer wieder interessant. Auf dem Amiga kann man auch mit Basic in vertretbaren Zeiten solche Grafiken erstellen lassen.

Hat man eine Funktion, die von zwei Variablen abhängig ist, kann man diese dreidimensional darstellen. Da der Bildschirm aber nur zwei Dimensionen besitzt, muß man die Räumlichkeit durch geschickte Umrechnung der Koordinaten erreichen.

Bei dem vorliegenden Programm wird der räumliche Eindruck noch dadurch verstärkt, daß die einzelnen Flächen mit einer Farbe gefüllt werden. Die nicht sichtbaren Flächen und Linien werden somit verdeckt.

Nun eine kurze Erläuterung des Programms (Listing 1). Der erste Teil »EinlesenDefinitionsbereich« dient zur Eingabe der Funktionsgrenzen. »x« und »y« bestimmen die Fläche in der die Funktionswerte berechnet werden. Durch verschiedene Eingaben für die »z«-Werte können Sie die Ausdehnung der Grafik in z-Richtung festlegen.

Nach der Abfrage, ob die Eingabewerte korrekt sind, geht dann die Berechnung der Grafik los. Zuerst wird der Grafikbildschirm eingeschaltet, da das Programm die einzelnen Flächen sofort zeichnet.

Nach dem Errechnen der Schrittweiten für x- und y-Achse wird ein Feld »p%« mit Werten gefüllt, die später benötigt werden. Es folgt ein längerer Teil, in dem die Eckpunkte für das zu zeichnende Viereck berechnet werden.

Ab dem Label »Graphzeichnen« werden dann nur noch die Flächen gefüllt und andersfarbige Linien außenherum gezogen.

Wollen Sie eigene Funktionen darstellen lassen, müssen Sie nur in Zeile 94 die Formel zur Berechnung von z verändern. Hier hilft genau wie bei den Parametern nur das Ausprobieren von verschiedenen Werten.

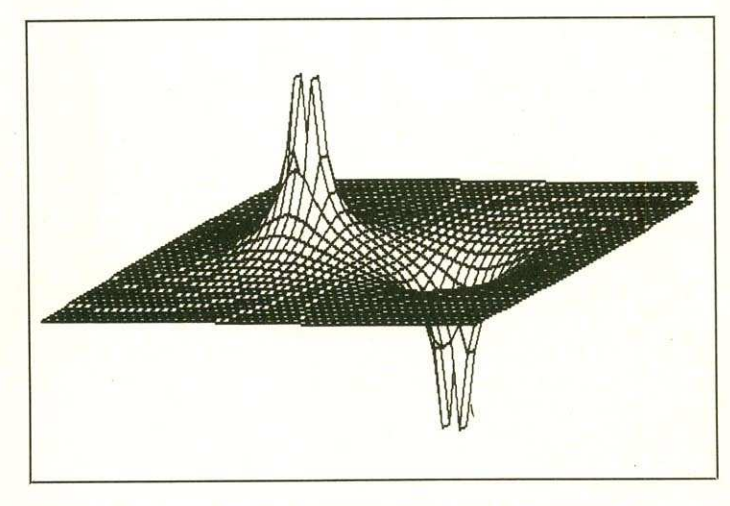


Bild 1. Eine schöne Beispielfunktion des Programms

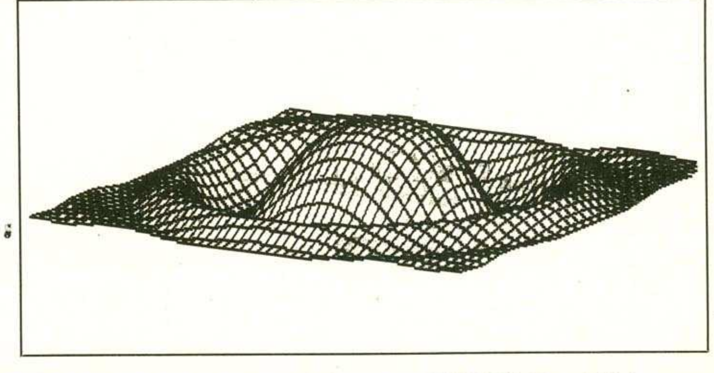


Bild 2. Die Funktion z=COS(x^2+y^2)/EXP((x^2+y^2)/4)

Nachfolgend finden Sie drei interessante Funktionen mit den passenden Werten, die Sie einmal ausprobieren sollten.

```
z=1/SQR((x+1)^2+y^2)-1/SQR((x-1)^2+y^2)
xmin : -3        xmax : 3        ymin : -3
ymax : 3         zmin : -5       zmax : 5
```

(Bild 1)

```
z=COS(x^2+y^2)/EXP((x^2+y^2)/4)
xmin : -3        xmax : 3        ymin : -3
ymax : 3         zmin : -2       zmax : 2
```

(Bild 2)

```
z=COS(x^2+y^2)/(SQR(ABS(x)+.1))
xmin : -3        xmax. : 3       ymin : -3
ymax : 0         zmin : -3       zmax : 3
```

Bei den eingegebenen Funktionen wird keine Überprüfung durchgeführt. Sie müssen deswegen bei Ihren Funktionen und Werten aufpassen, daß nicht Stellen enthalten sind, an denen die Funktionen unendliche Werte annehmen. Sie können zwar das Glück haben, daß durch die Schrittweite zwischen den berechneten Punkten genau die kritische Stelle nicht berechnet wird, aber der Graph sieht dann meistens nicht gut aus.

Und nun viel Spaß beim Experimentieren mit dem Programm.

(rb)

```
 1 '***************************
 2 '*     Netzgrafik Amiga     *
 3 '*           von            *
 4 '*      Gerald Heckner      *
 5 '*           und            *
 6 '*        Jörg Peters       *
 7 '***************************
 8
 9 EinlesenDefinitionsbereich:
10
11 INPUT "x Min";xl
12 INPUT "x Max";xr
13 INPUT "y Min";yh
14 INPUT "y Max";yv
15 INPUT "z Min";zu
16 INPUT "z Max";zo
17 PRINT "Eingabe Korrekt ? (j/n)"
18
19 Korrektheit:
20
21 eg$ = INKEY$
22 IF eg$ = "n" THEN GOTO EinlesenDefinitions-
   bereich
23 IF eg$ <> "j" THEN GOTO Korrektheit
24
25 Grafikbildschirmeinschalten:
26
27 SCREEN 2,640,400,2,2
28 WINDOW 2
29
30 Polyedereckpunkte:
31
32 dx = (xr-xl)/408
33 dy = (yv-yh)/66
34 zd = zo-zu:dz = zd/133:z0 = zo/dz
35 x1 = 231: x2 = 631
36 DIM p%(52)
37 i = 1
38 FOR a = x1 TO 639 STEP 8
39  x=xl+(a-231)*dx
40  y=yh
41  GOSUB Funktion
42  ac = INT(z0-z/dz)
43  p%(i) = ac
44  i=i+1
45 NEXT a
46 FOR b1 = z0 TO z0+64 STEP 2
47 i = 1
48  FOR a=x1 TO x2 STEP 8
49    b = p%(i)
50    c = a+8
51    d = p%(i+1)
52    g = a-7
53    x = xl+(a-x1)*dx
54    y = yh+(b1+2-z0)*dy
55    IF i > 1 THEN h = f: GOTO polyeder
56    GOSUB Funktion
57    h = INT(b1+2-z/dz)
58
59    polyeder:
60
61    p%(i) = h
62    e = a+1
63    x = xl+(a-x1+8)*dx
64    GOSUB Funktion
65    f = INT(b1+2-z/dz)
66    IF i = 51 THEN p%(52) = f
67    IF a>617 OR a<0 OR c>617 OR c<0  OR e
      >617 OR e<0 OR g>619 OR g<0 THE N
      SEnde
68    IF b>186 OR b<0 OR d>186 OR d<0  OR f
      >186 OR f<0 OR h>186 OR h<0 THE N
       SEnde
69
70    Graphzeichnen:
71
72    AREA (a,b)
73    AREA (c,d)
74    AREA (e,f)
75    AREA (g,h)
76    AREAFILL
77
78    LINE (a,b)-(c,d),2
79    LINE (c,d)-(e,f),2
80    LINE (e,f)-(g,h),2
81    LINE (g,h)-(a,b),2
82
83    SEnde:
84
85    i = i+1
86    NEXT a
87
88 x1 = x1-7: x2 = x2-7
89 NEXT b1
90
91 END
92
93 Funktion:
94
95 REM **** F(x/y) = ***
96 z=COS(x^2+y^2)/EXP((x^2+y^2)/4)
97 RETURN
```

Listing 1. Das komplette Listing zu Netzgrafik

AMIGA SOFTWARE

Precision Software

Superbase

Relationales Datenbank-System für den Amiga mit mindestens 512 Kbyte RAM in deutscher Sprache

Superbase – das relationale Datenbank-System

Superbase vereint als erstes Programm einer neuen Generation von Datenbank-Systemen sowohl eine neuartige, äußerst benutzerfreundliche Bedienung mit Pull-down-Menüs, Fenstern und Maussteuerung als auch die enorme Leistungsfähigkeit einer relationalen Dateiverwaltung.

Einfacher Datenbank-Aufbau

Mit den leichtverständlichen Pull-down-Menüs und Kontrollfeldern legen Sie in Minuten eine komplette Datenbank an. Sie können ein bereits festgelegtes Format jederzeit ändern, ohne Ihre Daten zu zerstören.

Verwaltung der Daten

Superbase zeigt Ihre Daten auf verschiedene Arten an, beispielsweise als Tabelle oder als Formular. Sind Index und Felder selektiert, so können Sie Ihre Daten wie bei einem Videorecorder anzeigen lassen. Schneller Vorlauf, Rücklauf, Pause und Stop – ein Recorder ist nicht einfacher zu bedienen. Ein einzigartiges Filtersystem wählt beliebige Datenkategorien aus, mit denen Sie dann arbeiten können.

Die Stärken von Superbase

Das Festlegen von Übersichten und zusammenhängenden Abfragen über mehrere verknüpfte Dateien ist auch bei verschiedenen Sortierkriterien kein Problem. Daten anderer Datenbanken oder Anwenderprogramme lassen sich ebenfalls problemlos verarbeiten. Binden Sie Daten in Ihre Textverarbeitung ein oder bilden Sie aus verschiedenen Dateien eine neue Datenbank! Die fortschrittliche Baumstruktur und die Disketten-Pufferung garantieren immer höchste Leistungsfähigkeit – Superbase findet beispielsweise einen bestimmten Datensatz in einer Datei, die 100 Adressen umfaßt, in nur 0,5 Sekunden.

Datenbank mit Bildern

Superbase bietet neben den gängigen Datenbank-Funktionen die Möglichkeit, Bilder und Grafiken darzustellen und zu verwalten. Einzigartigen Grafik-Datenbanken oder Dia-Shows steht also nichts im Wege.

Wer braucht Superbase?

Die Anwendungsmöglichkeiten sind nahezu unbegrenzt.

Hier einige Beispiele:

Geschäftliches	Professionelle Anwendungen
Lagerbestand	Design
Fakturierung	Fotografie
Registratur	Journalismus
Versandlisten	Sammlungen
Verwaltung	Forschung
Adressen	Ausbildung

Leistungsumfang

Die Software: ● bis zu 17 Gigabyte Speicherkapazität pro Datei ● bis zu 16 Millionen Datensätze pro Datei ● maximal 999 Indizes pro Datei ● Anzahl der geöffneten Dateien, Anzahl der Dateien und Anzahl der Felder pro Datensatz: jeweils systemabhängig. Zum Beispiel: Für eine übliche Adreßverwaltung bei einer Datensatzlänge von 200 Byte können Sie auf Ihrer Diskette (880 Kbyte freier Speicher) ca. 4000 Adressen speichern.

Die Daten: ● Text, Daten, numerische Felder und externe Dateien ● Überprüfung bei der Eingabe ● Formelfelder ● Kalender der Jahre 1–9999, verschiedene Datumsformen ● verschiedene Zahlenformate bei 13stelliger Genauigkeit ● Datenschutz per Paßwort

Die Ausgaben: ● das Programm beherrscht einen flexiblen Etikettendruck und produziert übersichtliche Listen mit dem Reportgenerator ● bis zu 255 Spalten ● mit Titel, Datum und Seitenzahl ● Datensatz-Zähler, Durchschnitt, Zwischen- und Endergebnis ● Ausgabe von mehreren Dateien auf Bildschirm, Drucker, Diskette oder neuer Datei ● mehrspaltiger Etikettendruck mit variablem Format ● Speicherung der Ausgabe- und Abfrage-Formate zur späteren Verwendung ● vielfältige Sortierkriterien

Hardware-Anforderung

Amiga mit mindestens 512 Kbyte RAM, beliebiger Drucker mit Centronics-Schnittstelle.

Best.-Nr. 51636

DM 249,–* (sFr 199,–/öS 2490,–*)

* inkl. MwSt. Unverbindliche Preisempfehlung

Markt&Technik-Produkte erhalten Sie in den Fachabteilungen der Warenhäuser, im Versandhandel, in Computerfachgeschäften oder direkt beim Verlag. Fragen Sie nach dem Gesamtverzeichnis Frühjahr '87.



Markt&Technik Verlag AG, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH, Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

702226

Ein Sprite- und Bob-Editor mit Extras

Um Sprites und Bobs (bewegliche Grafiken) zu erstellen, benötigt man einen Editor. Wenn er noch etwas mehr kann, als nur Grafiken erstellen, ist es ein gutes Hilfsmittel für jeden Programmierer.

Bobs und Sprites sind grafische Gebilde, die programmgesteuert unabhängig von bestehenden Bildern oder Grafiken über den Bildschirm huschen können. Um diese Gebilde erstellen zu können, benötigt man einen Editor für Sprites und Bobs.

Dem Amiga wird auf der Extra-Diskette in der Schublade »BasicDemos« bereits ein Objekteditor namens »ObjEdit« mitgeliefert. Allerdings ist dieses Programm für viele Bereiche völlig ungenügend. Zu bemängeln ist vor allem, daß

1. nur mit zwei Bitebenen, also mit vier Farben, gearbeitet werden kann. Um mehr Bitebenen zu bekommen, müßte jeweils das Programm umgeschrieben werden.
2. sich die Farben innerhalb des Programms nicht ändern lassen. Dazu müßte man erst wieder »Preferences« von der Workbench-Diskette laden, die Farben ändern und wieder zum »ObjEdit« zurückkehren. Das alles ist viel zu umständlich.
3. jeweils nur ein Objekt (Bob oder Sprite) erzeugt werden kann. Für Bewegungsabläufe (etwa ein fliegender Vogel) müssen verschiedene Objekte gleichzeitig darstellbar sein, damit man die Unterschiede zwischen den einzelnen Bewegungsphasen genau feststellen kann.
4. sich nur eine Auflösungsstufe verwenden läßt: 640 x 200 Punkte. Oft möchte man jedoch auch mit einer Auflösung von 320 x 200 Punkten arbeiten, was für viele Spiele auch genügt und außerdem Speicherplatz spart.
5. der »ObjEdit« sich nur für grobe Formen eignet. Gerade bei größeren Objekten, bei denen die Vergrößerungsfunktion versagt, können keine Details gezeichnet werden.
6. man keine Informationen über die Zahlenwerte, aus denen sich das Objekt zusammensetzt, erhält.

Dagegen arbeitet der neue, in Amiga-Basic geschriebene »ObjektEditor« (Listing) mit wahlweise 1 bis 4 Bitebenen und 2 bis 16 Farben in den Auflösungen 320 x 200 oder 640 x 200 Bildpunkte. Es können damit gleichzeitig sieben Sprites und bis zu 32 Bobs erzeugt werden, die sich zudem noch miteinander kombinieren lassen. Eine Änderung der Farben ist jederzeit möglich.

Programmbedienung

Nach dem Start meldet sich der Objekteditor mit einem kleinen Menü, um die Zahl der Bitebenen und die Auflösung einzustellen. Klicken Sie zunächst mit der linken Maustaste entweder »Sprites« oder »Bobs« an. Sprites arbeiten immer mit zwei Bitebenen, also vier Farben. Bei Bobs muß zusätzlich angegeben werden, wie viele Farben gleichzeitig dargestellt werden sollen. Anschließend ist noch die Auflösung (320 x 200 oder 640 x 200) zu bestimmen.

Man befindet sich nun im eigentlichen Editor. Klicken Sie zunächst eine Zahl in der oberen Mitte des Bildschirms an. Dies ist die Nummer des aktuellen Objekts, mit dem Sie arbeiten. Bei Bobs können Sie aus maximal 32 Objekten auswählen, bei Sprites aus maximal sieben, da von der möglichen Zahl von acht Sprites noch der Mauszeiger (ein Sprite) abgezogen werden muß.

Die aktuelle Objektnummer zeigt das Programm gesondert an. Links sehen Sie das Editorfeld mit einem eigenen Cursor. Der Editor-Cursor läßt sich mit den Cursortasten innerhalb des Editorfeldes bewegen.

Um mit dem Cursor zu zeichnen, ist die Leertaste zu drücken. Wenn Sie jetzt den Cursor bewegen, zeichnet er solange in der aktuellen Farbe, bis die Leertaste wieder gedrückt wird.

Klicken Sie nun wieder auf Ihr aktuelles Objekt, und es erscheint sofort in seiner richtigen Gestalt.

Voreingestellt ist die Farbe »1«, ein Grauwert. Wenn mit einer anderen Farbe gearbeitet werden soll, kann im Editor direkt eine andere Farbe angeklickt werden. Es ändert sich damit die Farbnummer und der Cursor arbeitet mit der neu gewählten Farbe. Farbe »0« ist immer »durchsichtig«, also die Hintergrundfarbe.

Die sichtbaren Farben können auch geändert werden. In der unteren Mitte des Bildschirms sehen Sie ein Rechteck mit dem Zusatz »Farben ändern«. »R-G-B« steht für den Rot-, Grün- und Blauanteil der Farbe. Klicken Sie nun in eines der Felder, beispielsweise das Feld mit dem Buchstaben »R«. Mit <+> und <-> kann man nun den Rotanteil der aktuellen Farbe ändern. Sind Sie mit dem Ergebnis zufrieden, drücken Sie die Leertaste. Der Objekteditor arbeitet mit dem neuen Farbwert weiter. Das gleiche können Sie mit jeder anderen Farbe zu jedem Zeitpunkt machen.

Noch ein Wort zur Anzeige der Farbwerte: Von Amiga-Basic aus lassen sich die Farben mit dem PALETTE-Befehl ändern. Wie Sie vermutlich wissen, müssen bei PALETTE die Farbwerte als Zahlen zwischen »0« und »1« eingegeben werden. Diese Zahlen ergeben sich aus der Division der Zahlen 0 bis 15 durch 15. Es ist aber einfacher, die ganzen Zahlen vor sich zu sehen, als die Brüche, wie sie beispielsweise im Amiga-Basic-Handbuch aufgeführt sind.

Zusätzlich findet der Anwender in der Menüzeile noch einige Sonderfunktionen:

»End« und »Quit« beenden die Arbeit mit dem Objekteditor. Das Programm sollte nur beendet werden, wenn Sie alle Objekte oder Sprites vorher gespeichert haben, sonst sind Sie verloren.

»Run« startet das Programm neu und beginnt wieder am Anfang mit der Abfrage nach Bob oder Sprite.

Unter dem Menüpunkt »Dateien« finden sich die Unterpunkte »Laden«, »Speichern« und »Drucken«. Soll ein Objekt geladen werden, ist zuerst die Positionsnummer anzugeben, an der das Objekt erscheinen soll. Nun kann das Objekt geladen werden. Das Programm fordert Sie auf, den Dateinamen anzugeben. Nach <RETURN> wird das Objekt, wenn vorhanden, geladen.

Genauso funktioniert »Speichern«: Erst die gewünschte Objektnummer angeben und anschließend unter einem neuen Namen speichern. Die Objekte werden übrigens automatisch mit zwei Zusätzen gespeichert, die Sie nicht mit eingeben dürfen. Haben Sie beispielsweise ein Sprite mit dem Namen »Stern« gespeichert, besitzt es den Directory-Namen »Stern.Spr«. Bobs haben den Zusatz ».Bob«. Außerdem tragen Bobs noch einen weiteren Zusatz, nämlich die Zahl der Bitebenen. Ein gespeichertes Objekt mit dem Namen »Stern.Bob4« bedeutet: Das Objekt ist ein Bob mit vier Bitebenen, besitzt also 16 Farben. Bobs können nur geladen werden, wenn ihre Bitebenen-Anzahl mit der entsprechenden Bitebenen-Einstellung des Objekteditors übereinstimmen. Bei Sprites müssen keine Bitebeben angegeben werden. Sie sind überall einsetzbar und tragen ihre Farbinformationen gewissermaßen im Huckepack bei sich.

Der Menüpunkt »Drucken« gibt Ihnen die Zahlenwerte aus, aus denen das Objekt zusammengesetzt ist. Bei Sprites gibt das Programm zusätzlich die Farbwerte aus. Vermutlich werden Sie aufgefordert, die Workbench-Diskette einzulegen, damit der Druckertreiber geladen werden kann. Findet der Amiga nicht zum Objekteditor zurück, müssen Sie im Menü »Windows« den Unterpunkt »Show Output« anklicken oder den Screen nach unten ziehen. Die ausgedruckten Werte können Sie vielleicht gebrauchen, wenn Sie selber abtippbare Programme erstellen wollen.

Der Objekteditor erzeugt für jedes Objekt einen String. Dies geschieht bei der Initialisierung des Programms (Unterprogramm »preobstring« = Objektstring vorbereiten) und bei jeder Eingabe im Editorfeld. Dies ist auch der Grund dafür, warum der Cursor bei vier Bitebenen etwas »nachhinkt«. Er muß ja für jeden neuen Bildpunkt pro Bitebene einen Wert an die richtige Stelle schreiben.

»Objektlänge« ist nicht im Objektstring enthalten und soll Sie nur über die Länge informieren. »ColorSetOffset«, »DataSetOffset«, »Anzahl der Bitebenen«, »Objektbreite« und »Objekthöhe« sind jeweils Long-Integer-Zahlen und nehmen daher im Objektstring die Länge 4 ein. Die anderen Werte sind Short-Integer-Zahlen der Stringlänge 2. Sie finden dann nach Bitebenen geordnet die jeweiligen Zahlenwerte, aus denen sich das Objekt zusammensetzt.

Bei Sprites finden Sie am Schluß die Sprite-eigenen Farbwerte. Die jeweils erste Zahl steht folgendermaßen im Objektstring: Sie errechnet sich aus 256 * Rotanteil + 16 * Grünanteil + Blauanteil einer Farbe. Es handelt sich um die Farbwerte der Farben 1 bis 3, aus denen das Sprite erstellt wurde.

Im Menü »Ed-Funktionen« sind einige Hilfen zur Erstellung der Objekte enthalten. »Backcol« bedeutet Hintergrundfarbe setzen. Wenn Sie beispielsweise wissen, daß Ihr Objekt vor allem aus einer Farbe besteht, dann sollten Sie einfach diese Farbe einstellen, sie aktualisieren, die jeweilige Objektnummer und anschließend »Backcol« anklicken. Das Programm erzeugt ein Objekt, das zunächst nur aus dieser Farbe besteht. Sie ersparen sich damit Zeit, da das Objekt nicht mehr mit dem Cursor ausgefüllt werden muß.

»Obback« bringt ein Objekt ins Editierfeld zurück. Haben Sie bereits ein Objekt erzeugt, dann klicken Sie es an und wählen »Obback«. Es wird nun Pixel für Pixel ins Editierfeld zurückgebracht, wo es erneut verändert werden kann. Diese Funktion ist leider langsam, da Amiga-Basic hier nicht besonders schnell ist.

»Obclear« vernichtet das aktuelle Objekt und löscht das Editierfeld.

»Farbreset« bringt die ursprüngliche Farbeinstellung zurück. Sollten Sie während der Arbeit an den Objekten die Farben geändert haben, werden Sie damit zurückgesetzt. Es sind 16 Farben fest eingestellt. Sprites behalten allerdings ihre vorherigen Farben. Um die Farbe fertiger Sprites zu ändern, muß der Umweg über »Farben ändern« genommen werden.

Schließlich existieren noch die beiden Funktionen »Short« und »Obcomb«. Diese Funktionen dürfen erst am Ende Ihrer Arbeit verwendet werden.

Mit »Short« lassen sich Objekte in vertikaler Richtung kürzen, indem das untere Ende des Objekts abgeschnitten wird. Dadurch können Sie in einem Programm eventuell Speicherplatz und Rechenzeit sparen. Eine Kürzung in horizontaler Richtung wäre Unsinnig, deshalb ist auch keine Funktion dafür vorgesehen.

Mit der Funktion »Obcomp« lassen sich nun die einzelnen Objektteile zu größeren Objekten »zusammenschweißen«. Das Objekt-Anzeigefeld wurde so konzipiert, daß die einzelnen Teile nahtlos neben- und übereinander stehen. So läßt sich feststellen, ob die Teile zusammenpassen.

Die »Obcomp«-Funktion arbeitet in zwei Richtungen. Die Teile lassen sich in vertikaler und horizontaler Richtung zusammenfassen. Dabei muß allerdings eine feste Reihenfolge eingehalten werden:

— Sprites lassen sich nur in vertikaler Richtung zusammensetzen. Bestrebungen, dies auch in horizontaler Richtung zu tun, ignoriert der Objekteditor. Am besten kürzen Sie (falls nötig) den Teil, der unten sein soll und wählen dann »Obcomp«. Geben Sie nun <v> für vertikal ein und danach die beiden Teile, die verbunden werden sollen. Das kombinierte Objekt hat immer die Nummer, die Sie als erste eingeben. Die linke obere Ecke dieses Teils spricht man mit OBJECT.X- und OBJECT.Y-Befehlen an. Sprites können also maximal 16 Punkte breit, aber beliebig lang sein.

— Bobs dagegen lassen sich in beide Richtungen kombinieren. Beachten Sie, daß immer eine gleichmäßige rechtwinklige Form entstehen muß.

Als erstes kürzen Sie (wenn nötig) alle unteren Enden mit »Short«. Danach setzen Sie alle Teile zuerst in vertikaler und anschließend in horizontaler Richtung zusammen. Andernfalls riskieren Sie einen Systemabsturz mit Verlust aller Daten. Speichern Sie schließlich das erzeugte Objekt auf Diskette.

Sie können mit »Obcomp« ein Teil auch verdoppeln. Sprite 1 und Sprite 1 in vertikaler Richtung ergibt also 2mal Sprite 1 übereinander mit der Nummer 1.

Mit »Obcomp« erzeugte Objekte lassen sich übrigens nicht mehr mit »Obclear« löschen. Laden Sie einfach an die jeweilige Objektnummer ein einzelnes Objekt. Dadurch verschwindet das zusammengesetzte Objekt. (Rainer Witschel/dm)

```
1
2
3
4 anfang:
5  CLEAR ,40000&
6  DEFINT a-z:DIM ob$ (32),fw(15,2)
7  LINE (60,25)-(182,45),3,b
8  LOCATE 5,10:PRINT"Objekteditor"
9  LOCATE 11,10:PRINT"Sprites"
10 LINE (60,73)-(150,93),2,b
11 LOCATE 14,10:PRINT"Bobs"
12 LINE (60,97)-(150,117),2,b
13 LOCATE 14,27:PRINT"2 Farben     4 Farben     8 Farben     16 Farben"
14 FOR i=0 TO 3
15  LINE (200+i*104,97)-(200+i*104+80,117),1,b
16 NEXT
17 LOCATE 17,10:PRINT"Auflvsung"
18 LINE (60,121)-(150,141),2,b
19 LOCATE 17,27:PRINT"320*200      640*200"
20 FOR i=0 TO 1
21  LINE (200+i*104,121)-(200+i*104+80,141),1,b
22 NEXT
23
24 auswahl:
25 test=MOUSE(0)
26 ON MOUSE GOSUB mausauswertung
27 MOUSE ON
28 LOCATE 8,10:PRINT"Waehlen Sie aus!"
29 FOR i=1 TO 2000:NEXT
30 LOCATE 8,10:PRINT SPACE$(16)
31 FOR i=1 TO 2000:NEXT
32 GOTO auswahl
33
34 mausauswertung:
35 test=MOUSE(0)
36 xx=MOUSE(1):yy=MOUSE(2)
37 IF xx>60 AND xx<150 THEN
38  IF yy>73 AND yy<93 THEN modus=1:bitplanes&=2:LINE (60,73)-(150,93),
   3,b
39  IF yy>97 AND yy<117 THEN modus=2:LINE (60,97)-(150,117),3,b
40  IF yy>121 AND yy<141 THEN aufloesung=1:LINE (60,121)-(150,141),3,b
41 END IF
42 IF yy>97 AND yy<117 AND xx>200 AND xx<592 AND modus=2 THEN
43  bitplanes&=INT(xx-96)/104:i=bitplanes&-1
44  LINE (200+i*104,97)-(200+i*104+80,117),3,b
45 END IF
46  IF yy>121 AND yy<141 AND xx>200 AND xx<384 AND aufloesung=1 THEN
47   auf=(INT(xx-96)/104):aufloesung=auf*320
48   LINE(200+(auf-1)*104,121)-(200+(auf-1)*104+80,141),3,b
49  END IF
50 IF modus=2 AND bitplanes&<>0 AND auf<>0 THEN GOTO startover
51 IF modus=1 AND auf <>0 THEN GOTO startover
52 GOTO auswahl
53
54 startover:
55 SCREEN 1,aufloesung,200,bitplanes&,auf
56 WINDOW 1,"Objekteditor",,,1
57 MENU 1,0,1,"END":MENU 1,1,1,"end":MENU 1,2,1,"quit":MENU 1,3,1,"run"
58 MENU 2,0,1,"Dateien":MENU 2,1,1,"laden"
59 MENU 2,2,1,"speichern":MENU 2,3,1,"drucken"
60 MENU 3,0,1,"Ed-Funktionen":MENU 3,1,1,"backcol":MENU 3,2,1,"obback"
61 MENU 3,3,1,"obclear":MENU 3,4,1,"obcomb":MENU 3,5,1,"farbreset"
62 MENU 3,6,1,"short"
63 IF auf=1 THEN auf=2 ELSE auf=1
64  IF modus=1 AND auf=1 THEN GOSUB prespriteh
65  IF modus=2 AND auf=1 THEN GOSUB prebobh
66  IF modus=1 AND auf=2 THEN GOSUB prespritel
67  IF modus=2 AND auf=2 THEN GOSUB prebobl
68  LINE (28/auf,14)-(162/auf,16+obfeldy*edy),1,b
69  LOCATE 4,56/auf:PRINT"Farbe"
70  colz=2^bitplanes&-1
71  FOR col=0 TO colz
72   LINE (520/auf,14+col*8)-(600/auf,14+col*8+7),col,bf
73  NEXT
74  LINE (520/auf,13)-(600/auf,21),1,b
75  LOCATE 17,24/auf:PRINT"Farbe":LOCATE 18,24/auf:PRINT"Aendern"
76  LOCATE 17,38/auf:PRINT"R"SPACE$(2/auf)"-"SPACE$(4/auf)"0"SPACE$(4/auf
    )"+"
77  LOCATE 19,38/auf:PRINT"G"SPACE$(2/auf)"-"SPACE$(4/auf)"0"SPACE$(4/auf
    )"+"
78  LOCATE 21,38/auf:PRINT"B"SPACE$(2/auf)"-"SPACE$(4/auf)"0"SPACE$(4/auf
    )"+"
79  FOR i=0 TO 2
80   LINE (280/auf,124+i*16)-(440/auf,124+i*16+16),1,b
81  NEXT
82  x=32/auf:y=16:altx=32/auf:alty=16:biteb=bitplanes&-1
83  LINE (x-1,y-1)-(x+edx-1,y+edy-1),1,b
84  maxy=16+(obfeldy-1)*edy
85  GOSUB farbreset
86  GOSUB preobstring
87  GOTO bewegen
88
89 prespriteh:
90  edx=8:edy=5:obfeldx=32:obfeldzx=4:obfeldy=32:obfeldzy=2
```

Listing. Ein Sprite- und Bob-Editor für den Amiga

```
91  LOCATE 4,24:PRINT"Sprite"
92  LOCATE 7,2:PRINT"8-":LOCATE 12,1:PRINT"16-"
93  LOCATE 17,1:PRINT"24-":LOCATE 22,1:PRINT"32-"
94  LOCATE 5,36:PRINT" 1  2  3  4"
95  LOCATE 9,36:PRINT" 5  6  7  8"
96  RETURN
97
98 prebobh:
99  edx=8:edy=10:obfeldx=16:obfeldzx=8:obfeldy=16:obfeldzy=4
100  LOCATE 4,25:PRINT"Bob"
101  LOCATE 12,2:PRINT"8-":LOCATE 22,1:PRINT"16-"
102  LOCATE 3,36:PRINT" 1   3   5   7"
103  LOCATE 4,36:PRINT"   2   4   6   8"
104  LOCATE 5,36:PRINT" 9  11  13  15"
105  LOCATE 6,36:PRINT"  10  12  14  16"
106  LOCATE 7,36:PRINT"17  19  21  23"
107  LOCATE 8,36:PRINT"  18  20  22  24"
108  LOCATE 9,36:PRINT"25  27  29  31"
109  LOCATE 10,36:PRINT"  26  28  30  32"
110  RETURN
111
112 prespritel:
113  edx=4:obfeldx=16:obfeldzx=4:obfeldy=32:obfeldzy=2:edy=5
114  LOCATE 4,12:PRINT"Sprite"
115  LOCATE 7,2:PRINT"8":LOCATE 12,1:PRINT"16"
116  LOCATE 17,1:PRINT"24":LOCATE 22,1:PRINT"32"
117  LOCATE 5,18:PRINT" 1 2 3 4"
118  LOCATE 9,18:PRINT" 5 6 7 8"
119  RETURN
120
121 prebobl:
122  edx=4:edy=5:obfeldx=16:obfeldzx=4:obfeldy=32:obfeldzy=2
123  LOCATE 4,12:PRINT"Bob"
124  LOCATE 7,2:PRINT"8":LOCATE 12,1:PRINT"16"
125  LOCATE 17,1:PRINT"24":LOCATE 22,1:PRINT"32"
126  LOCATE 5,18:PRINT" 1 2 3 4"
127  LOCATE 9,18:PRINT" 5 6 7 8"
128  RETURN
129
130 preobstring:
131  IF modus=1 THEN sprbit=1 ELSE sprbit=0
132  colorsetoffset&=0:datasetoffset&=0:breite&=16
133  shortflags=24+sprbit:planepick=0:planeonoff=0
134  farbe1=3276:farbe2=885:farbe3=3888:hoehe&=obfeldy
135  laenge=26+bitplanes&*hoehe&*2+sprbit*6
136  teilz=obfeldzy*obfeldzx:biteb=bitplanes&-1
137  FOR i=1 TO teilz
138   ob$(i)=STRING$(laenge,0)
139   MID$(ob$(i),1,4)=MKL$(colorsetoffset&)
140   MID$(ob$(i),5,4)=MKL$(datasetoffset&)
141   MID$(ob$(i),9,4)=MKL$(bitplanes&)
142   MID$(ob$(i),13,4)=MKL$(breite&)
143   MID$(ob$(i),17,4)=MKL$(hoehe&)
144   MID$(ob$(i),21,2)=MKI$(shortflags)
145   MID$(ob$(i),23,2)=MKI$(planepick)
146   MID$(ob$(i),25,2)=MKI$(planeonoff)
147   IF modus=1 THEN
148    MID$(ob$(i),laenge-5,2)=MKI$(farbe1)
149    MID$(ob$(i),laenge-3,2)=MKI$(farbe2)
150    MID$(ob$(i),laenge-1,2)=MKI$(farbe3)
151   END IF
152  NEXT
153  RETURN
154
155 bewegen:
156  MENU ON:ON MENU GOSUB checkmenu
157  ON ERROR GOTO fehler
158  test=MOUSE(0)
159  ON MOUSE GOSUB maus
160  MOUSE ON
161  a$=INKEY$
162  IF a$<>""THEN
163   IF a$=CHR$(29) AND y<maxy THEN y=y+edy:GOTO cursorsetzen
164   IF a$=CHR$(31) AND x>32/auf THEN x=x-edx:GOTO cursorsetzen
165   IF a$=CHR$(30) AND x<152/auf THEN x=x+edx:GOTO cursorsetzen
166   IF a$=CHR$(28) AND y>16 THEN y=y-edy:GOTO cursorsetzen
167   IF a$=CHR$(32) THEN
168    IF sch=0 THEN sch=1 ELSE sch=0
169   END IF
170  END IF
171  IF sch=1 THEN GOTO fuellen
172  GOTO bewegen
173
174 cursorsetzen:
175  LINE (altx-1,alty-1)-(altx+edx-1,alty+edy-1),0,b
176  LINE (x-1,y-1)-(x+edx-1,y+edy-1),1,b
177  altx=x:alty=y
178  GOTO bewegen
179
180 fuellen:
181  wertx=x*auf/8-4:werty=(y-16)/edy
182  wert&=2^(15-wertx)
183  IF wertx=0 THEN wert&=-wert&
184  LINE (x,y)-(x+edx-2,y+edy-2),col,bf
185  FOR i=0 TO biteb
186   teil$(i)=MID$(ob$(nummer),27+hoehe&*i*2+werty*2,2)
187   teil(i)=CVI(teil$(i))
188   IF (col AND 2^i)=2^i THEN teil(i)=teil(i) OR wert&
189   IF (col AND 2^i)<>2^i AND (teil(i) AND wert&)=wert& THEN teil(i)=
    teil(i) XOR wert&
190   teil$(i)=MKI$(teil(i))
191   MID$(ob$(nummer),27+hoehe&*i*2+werty*2,2)=teil$(i)
192  NEXT
193  GOTO bewegen
194
195 checkmenu:
196  m0=MENU(0):m1=MENU(1)
197  IF m0=1 THEN ON m1 GOSUB beenden,verlassen,starten
198  IF m0=2 THEN ON m1 GOSUB laden,speichern,drucken
199  IF m0=3 THEN ON m1 GOSUB backcol,obback,obclear,obcomb,farbreset,short
200  RETURN
201
202 fehler:
203  IF ERR=53 THEN f$="file not found"
204  IF ERR=61 THEN f$="disk full"
205  IF ERR=64 THEN f$="bad file name"
206  LOCATE 1,2:PRINT f$:BEEP
207  FOR i=1 TO 3000:NEXT:LOCATE 1,2:PRINT SPACE$(40)
208  RESUME bewegen
209
210 maus:
211  test=MOUSE(0)
212  xx=MOUSE(1):yy=MOUSE(2)
213  IF xx>280/auf AND xx<412/auf AND yy>16 AND yy<80 THEN GOSUB
    teilewahl
214  IF xx>280/auf AND xx<440/auf AND yy>125 AND yy<173 THEN GOSUB
    farbe aendern
215  IF xx>524/auf AND xx<604/auf AND yy>13 AND yy<(16+2^bitplanes&*8)
    THEN GOSUB farbauswahl
216  RETURN
217
218
219 teilewahl:
220  nummer=INT((xx-284/auf)/obfeldx)+1+INT((yy-16)/obfeldy)*obfeldzx
221  LOCATE 6,24/auf:PRINT nummer
222  GOSUB ob
223  RETURN
224
225
226  farbeaendern:
227  rgb=INT((yy-125)/16)
228  LOCATE 15,37/auf:PRINT"space=zurueck"
229  a$=INKEY$
230   IF a$="+" AND fw(col,rgb)<15 THEN fw(col,rgb)=fw(col,rgb)+1
231   IF a$="-" AND fw(col,rgb)>0 THEN fw(col,rgb)=fw(col,rgb)-1
232   IF a$=" " THEN LOCATE 15,37/auf:PRINT SPACE$(14):RETURN
233   IF a$="" THEN GOTO farbeaendern
234   LOCATE 17+2*rgb,44/auf:PRINT fw(col,rgb)
235   FOR i=0 TO 2
236    pf!(i)=fw(col,i)/15
237   NEXT
238   PALETTE col,pf!(0),pf!(1),pf!(2)
239   IF col>0 AND modus=1 THEN
240    spritefarbe=256*fw(col,0)+16*fw(col,1)+fw(col,2)
241    MID$(ob$(nummer),laenge-(3-col)*2-1,2)=MKI$(spritefarbe)
242   END IF
243   GOTO farbeaendern
244   RETURN
245
246 farbauswahl:
247  col=INT((yy-14)/8)
248  LOCATE 6,58/auf:PRINT col
249  FOR rgb=0 TO 2
250   LOCATE 17+2*rgb,44/auf:PRINT fw(col,rgb)
251  NEXT
252  RETURN
253
254 farbreset:
255  RESTORE
256  FOR i=0 TO colz
257   FOR ii=0 TO 2
258    READ fw(i,ii)
259  NEXT ii,i
260  DATA 0,0,1,12,12,12,3,7,5,15,3,0
261  DATA 1,6,2,7,4,2,14,12,0,5,5,15
262  DATA 14,8,2,8,11,14,14,10,12,3,1,0
263  DATA 7,9,2,2,1,5,7,1,8,15,15,13
264  FOR i=0 TO colz
265   PALETTE i,fw(i,0)/15,fw(i,1)/15,fw(i,2)/15
266  NEXT
267  FOR rgb=0 TO 2
268   LOCATE 17+2*rgb,44:PRINT fw(1,rgb)
269  NEXT
270  col=1:LOCATE 6,58/auf:PRINT col
271  RETURN
272
273 ob:
274  OBJECT.SHAPE nummer,ob$(nummer)
275  OBJECT.PLANES nummer,2^bitplanes&-1
276  obyy=FIX((nummer-1)/obfeldzx)
277  oby=15+obyy*obfeldy+sprbit*10
278  obx=280/auf+(nummer-1-obyy*obfeldzx)*obfeldx+sprbit*2
```

```
279  OBJECT.X nummer,obx:OBJECT.Y nummer,oby
280  OBJECT.ON nummer
281  RETURN
282
283 beenden:
284  MENU RESET:END
285
286 verlassen:
287  MENU RESET:NEW
288
289 starten:
290  RUN
291
292 laden:
293  eg$="Dateiname - ":GOSUB eingabe
294  IF sprbit THEN n$=n$+".spr" ELSE n$=n$+".bob"+CHR$(49+biteb)
295  OPEN n$ FOR INPUT AS 1
296  a$=INPUT$(LOF(1)-1,1)
297  CLOSE 1
298  ob$(nummer)=a$
299  GOSUB ob
300  RETURN
301
302 speichern:
303  eg$="Dateiname - ":GOSUB eingabe
304  IF sprbit THEN n$=n$+".spr" ELSE n$=n$+".bob"+CHR$(49+biteb)
305  OPEN n$ FOR OUTPUT AS 1
306  PRINT#1,ob$(nummer)
307  CLOSE 1
308  RETURN
309
310 backcol:
311  GOSUB obclear
312  hh=hoehe&*2
313  FOR i=0 TO biteb
314   IF(col AND 2^i)=2^i THEN
315    MID$(ob$(nummer),27+hh*i,hh)=STRING$(hh,255)
316   END IF
317  NEXT
318  FOR h=0 TO hoehe&-1
319   s1=32/auf:h4=h*edy+16:h6=(h+1)*edy+14
320   FOR s=0 TO 15
321   LINE(s1+edx*(15-s),h4)-(s1+edx*(16-s)-2,h6),col,bf
322  NEXT s,h
323  sch=0
324  RETURN
325
326 obback:
327  h1=hoehe&-1:s1=32/auf:h3=hoehe&*2
328  FOR h=0 TO h1
329   h4=h*edy+16:h6=(h+1)*edy+14:h7=edx*16-2:h2=h*2+27
330   FOR s=0 TO 15
331    cl=0:s2=edx*(15-s):h5=h7-edx*s
332     FOR i=0 TO biteb
333      teil$(i)=MID$(ob$(nummer),h3*i+h2,2)
334      teil(i)=CVI(teil$(i))
335      IF (teil(i) AND 2^s)=2^s THEN cl=cl+2^i
336     NEXT i
337     LINE (s1+s2,h4)-(s1+h5,h6),cl,bf
338  NEXT s,h
339  sch=0
340  RETURN
341
342 obclear:
343  l1=bitplanes&*hoehe&*2
344  MID$(ob$(nummer),27,l1)=STRING$(l1,0)
345  LINE (31/auf,15)-(160/auf,15+obfeldy*edy),0,bf
346  GOSUB ob
347  RETURN
348
349 obcomb:
350  eg$="(v)ertikal-(h)orizontal(Bobs) - ":GOSUB eingabe
351  IF n$="v" THEN GOTO vertikal
352  IF n$="h" AND sprbit=0 THEN GOTO horizontal
353  GOTO obcomb
354
355 obcombshow:
356  ob$(num1)=kopf$+ebenen$+farb$
357  sch=0:e$="":ebenen$=""
358  OBJECT.SHAPE num1,ob$(num1)
359  OBJECT.PLANES num1,2^bitplanes&-1
360  OBJECT.X num1,40/auf:OBJECT.Y num1,20
361  OBJECT.ON num1
362  RETURN
363
364 vertikal:
365  eg$="vertikal: 1. Teil - ":GOSUB eingabe
366  num1=VAL(n$):IF num1<1 OR num1>teilz THEN GOTO vertikal
367  eg$="vertikal: 2. Teil - ":GOSUB eingabe
368  num2=VAL(n$):IF num2<1 OR num2>teilz THEN GOTO vertikal
369  hoch1$=MID$(ob$(num1),17,4):hoch2$=MID$(ob$(num2),17,4)
370  hoch1&=CVL(hoch1$):hoch2&=CVL(hoch2$)
371  hoh1=hoch1&*2:hoh2=hoch2&*2
372  hoch&=hoch1&+hoch2&
373  hoch$=MKL$(hoch&)
374  MID$(ob$(num1),17,4)=hoch$
375  kopf$=LEFT$(ob$(num1),26)
376  IF sprbit THEN farb$=RIGHT$(ob$(num1),6) ELSE farb$=""
377  FOR i=0 TO biteb
378   e$=MID$(ob$(num1),27+hoh1*i,hoh1)+MID$(ob$(num2),27+hoh2*i,hoh2)
379   ebenen$=ebenen$+e$
380  NEXT
381  GOTO obcombshow
382
383 horizontal:
384  eg$="horizontal: 1. Teil - ":GOSUB eingabe
385  num1=VAL(n$):IF num1<1 OR num1>teilz THEN GOTO horizontal
386  eg$="horizontal: 2. Teil - ":GOSUB eingabe
387  num2=VAL(n$):IF num2<1 OR num2>teilz THEN GOTO horizontal
388  breit1$=MID$(ob$(num1),13,4):breit2$=MID$(ob$(num2),13,4)
389  breit1&=CVL(breit1$):breit2&=CVL(breit2$)
390  breit&=breit1&+breit2&
391  breit$=MKL$(breit&)
392  MID$(ob$(num1),13,4)=breit$
393  kopf$=LEFT$(ob$(num1),26)
394  hoch$=MID$(ob$(num1),17,4)
395  hoch&=CVL(hoch$)
396  ebenen$="":br1=breit1&/8:br2=breit2&/8
397  FOR i=0 TO biteb
398   FOR ii=0 TO hoch&-1
399    e$=MID$(ob$(num1),27+hoch&*br1*i+ii*br1,br1)+MID$(ob$(num2),27+hoch&
       *br2*i+ii*br2,br2)
400    ebenen$=ebenen$+e$
401  NEXT ii,i
402  GOTO obcombshow
403
404 drucken:
405  eg$="Nummer - ":GOSUB eingabe
406  nummer=VAL(n$):IF nummer<1 OR nummer>teilz THEN GOTO drucken
407  a$=ob$(nummer)
408  LPRINT"Objektlaenge"LEN(a$)
409  k$=MID$(a$,1,4):k&=CVL(k$):LPRINT"ColorSetOffset: "k&
410  k$=MID$(a$,5,4):k&=CVL(k$):LPRINT"DataSetOffset: "k&
411  k$=MID$(a$,9,4):k&=CVL(k$):LPRINT"Anzahl der Bitebenen: "k&;:eb=k&
412   LPRINT 2^k&" Farben"
413  k$=MID$(a$,13,4):k&=CVL(k$):LPRINT"Objektbreite in Pixels: "k&:br=k&
414  k$=MID$(a$,17,4):k&=CVL(k$):LPRINT"Objekthoehe in Pixels: "k&:hoe=k&
415  k$=MID$(a$,21,2):k=CVI(k$):LPRINT"ShortFlags: "k;
416   IF k AND 1 THEN LPRINT"Sprite" ELSE LPRINT"Bob"
417  k$=MID$(a$,23,2):k=CVI(k$):LPRINT"PlanePick: "k
418  k$=MID$(a$,25,2):k=CVI(k$):LPRINT"PlaneOnOff: "k
419  zz=br/16*hoe*2
420  FOR i=0 TO eb-1
421   LPRINT:LPRINT:LPRINT"Werte der Ebene "i+1":"
422   LPRINT:pz=0
423   FOR ii=0 TO zz-2 STEP 2
424    k$=MID$(a$,27+i*zz+ii,2)
425    k=CVI(k$)
426    pz=pz+1:IF pz=13 THEN pz=1:LPRINT
427    LPRINT k;
428  NEXT ii,i
429  LPRINT
430  IF sprbit=1 THEN
431   LPRINT:LPRINT"Farbwerte (R-G-B): "
432   FOR i=LEN(a$)-5 TO LEN(a$)-1 STEP 2
433    k$=MID$(a$,i,2):k=CVI(k$)
434    kr=FIX(k/256):kg=FIX((k-kr*256)/16):kb=k-kr*256-kg*16
435    LPRINT k"= "kr;kg;kb
436   NEXT
437  END IF
438  RETURN
439
440 short:
441  ebenen$=""
442  eg$="Nummer - ":GOSUB eingabe
443  nummer=VAL(n$):IF nummer<1 OR nummer>teilz THEN GOTO short
444  eg$="neue Hoehe - ":GOSUB eingabe
445  nhoch&=VAL(n$):IF nhoch&<1 OR nhoch&>hoch& THEN GOTO short
446  hoch$=MID$(ob$(nummer),17,4)
447  hoch&=CVL(hoch$)
448  nhoch$=MKL$(nhoch&)
449  MID$(ob$(nummer),17,4)=nhoch$
450  kopf$=LEFT$(ob$(nummer),26)
451  IF sprbit THEN farb$=RIGHT$(ob$(nummer),6) ELSE farb$=""
452  FOR i=0 TO biteb
453   e$=MID$(ob$(nummer),27+hoch&*2*i,nhoch&*2)
454   ebenen$=ebenen$+e$
455  NEXT
456  ob$(nummer)=kopf$+ebenen$+farb$
457  e$="":ebenen$=""
458  RETURN
459
460 eingabe:
461  n$="":LOCATE 1,2:PRINT eg$;
462  weiter:
463   ei$=INKEY$
464    IF ei$ <>""THEN
465     PRINT ei$;
466     IF ei$<>CHR$(13) THEN n$=n$+ei$
467     IF ei$=CHR$(13) THEN LOCATE 1,2:PRINT SPACE$(40):RETURN
468    END IF
469   GOTO weiter                                              (c) M&T
```

Terminal-Clock, die Stoppuhr

Die Überwachung der Verbindungsdauer mit einer Mailbox ist eine Domäne dieses vielseitigen und schnellen Utilities.

Wie oft ist es Ihnen schon passiert, daß wegen zehn Sekunden, die zu lange in einer Mailbox oder Datenbank verbracht wurden, eine neue 23-Pfennig-Einheit auf der nächsten Telefonrechnung stand? Für Amiga-Besitzer hat dieses Ärgernis jetzt ein Ende! Mit dem Programm Terminal-Clock (Listing 1) haben Sie die Verbindungsdauer immer im Blick und können so in aller Ruhe die Verbindung rechtzeitig beenden.

Die Stoppuhr erscheint nach dem Aufruf in der rechten, oberen Ecke des Bildschirms im Menübalken und harmoniert so mit den gebräuchlichen DFÜ-Programmen für den Amiga. Es ist natürlich problemlos möglich, die Uhr auf dem Bildschirm zu verschieben oder beliebig viele Uhren aufzurufen.

Mit der rechten Maustaste gelangen Sie in ein Pull Down-Menü mit den Punkten »Start«, »Stop« und »Null«. Mit »Start« beginnt die Uhr zu zählen, dieser Punkt sollte also angewählt werden, sobald die Verbindung hergestellt ist. Mit »Stop« wird die Uhr angehalten, mit »Null« zurückgestellt.

Das Programm ist vollständig in C geschrieben und muß mit dem Aztec-Compiler compiliert und dem entsprechenden Linker gelinkt werden. Achtung: Da Terminal-Clock die Systemzeitroutine von Aztec benutzt, kann nicht mit Lattice C compiliert werden. Mit nur 7 KByte ist das Programm ausgesprochen kurz — aber nützlich.

(Ralf Dittmer/pd)

Achtung: Zeilennummern nicht abtippen!

```
  1 /***********************************/
  2 /**                              **/
  3 /**        TerminalClock         **/
  4 /**                              **/
  5 /**        by Ralf Dittmer       **/
  6 /**                              **/
  7 /**      (c) Markt & Technik     **/
  8 /**                              **/
  9 /***********************************/
 10
 11 #include <exec/types.h>          /* Einbinden der ver-    */
 12 #include <graphics/gfx.h>        /* wendeten Strukturen   */
 13 #include <intuition/intuition.h> /*  und Definitionen     */
 14 #include <devices/timer.h>
 15 #include <time.h>
 16
 17 struct IntuitionBase *IntuitionBase;  /*   Definition     */
 18 struct GfxBase *GfxBase;              /*  der Struktur-   */
 19 struct Window *mywind, *OpenWindow(); /*    pointer       */
 20 struct IntuiMessage *Msg, *GetMsg();
 21 struct MsgPort *timerport = NULL, *CreatePort();
 22 struct timerequest timereq;
 23
 24
 25 /*Jetzt werden die Daten des kleinen Fensters festgelegt   */
 26
 27  struct NewWindow nw = {
 28  491, 0, 149, 10,    /*Die rechte obere Ecke des Windows    */
 29
 30  0, 1,              /*    Die Farben fuer das Fenster      */
 31
 32  CLOSEWINDOW   |   /* Es wird eine Msg gesendet, wenn      */
 33  MENUPICK      |   /* - das Closegadget angeklickt o.      */
 34  REFRESHWINDOW |   /* - ein Menue gewaehlt wird, das       */
 35  ACTIVEWINDOW,     /* - Fenster Erneuerung braucht u.      */
 36                    /* - wenn es aktiviert wird             */
 37
 38  WINDOWCLOSE   |   /* Es werden Gadgets fuer Close,        */
 39  WINDOWDRAG    |   /* Front und Back in das                */
 40  WINDOWDEPTH   |   /* Fenster eingebaut. Es kann frei      */
 41  SIMPLE_REFRESH |  /* auf dem Screen verschoben werden.    */
 42  ACTIVATE,         /* Das Programm muss sich um das        */
 43                    /* Refresh selbst kuemmern und wird     */
 44                    /* beim Oeffnen sofort aktiviert.       */
 45
 46  NULL, NULL,  /*Gadget und Image Structure werden nicht    */
 47  NULL,        /*verwendet, das Window hat keinen Title     */
 48
 49  NULL, NULL,  /*Screen ist Workbench, und BitMap normal     */
 50  0, 0, 0, 0,  /*        Es ist keine Sizing moeglich        */
 51  WBENCHSCREEN /* Screen Name ist WBENCHSCREEN               */
 52 };
 53
 54 /* Als naechstes werden die Texte, die im Menue er-        */
 55 /* scheinen sollen, definiert. Zeichenmode ist JAM2        */
 56
 57 struct IntuiText Clck1 = {0,1,JAM2,0,0,NULL,
 58                           (UBYTE *)" Start ",NULL};
 59 struct IntuiText Clck2 = {0,1,JAM2,0,0,NULL,
 60                           (UBYTE *)" Stop  ",NULL};
 61 struct IntuiText Clck3 = {0,1,JAM2,0,0,NULL,
 62                           (UBYTE *)" Null  ",NULL};
 63
 64 /* Festlegen der Daten fuer das Menue */
 65
 66 struct MenuItem Clck1It = {NULL,0,0,56,8, ITEMTEXT |
 67                           HIGHCOMP | ITEMENABLED
 68                           ,0,(APTR)&Clck1,NULL,0,NULL,0};
 69
 70 /*                                                          */
 71 /*                                                          */
 72
 73 struct MenuItem Clck2It = {&Clck1It,/*Zeiger auf naechs-   */
 74                                /* tes MenueItem            */
 75                           0,8, /* X- und Y-Wert, an         */
 76                                /* dem das Item beginnt     */
 77                           56,8,/* Laenge und Hoehe des      */
 78                                /* Items in Pixel           */
 79                           ITEMTEXT | HIGHCOMP,
 80                           /* Menuepunkt ist ein Text        */
 81                           /* und wird beim anwaehlen        */
 82                           /* invers dargestellt.            */
 83                           0, /*                             */
 84                           (APTR)&Clck2,/* Textadresse       */
 85                           NULL,NULL,NULL,
 86                           /* Kein HIGHIMAGE, kein           */
 87                           /* Amiga-Key, kein Untermen.      */
 88                           0 /* NextSelect = NULL            */
 89                           };
 90
 91 struct MenuItem Clck3It = {&Clck2It,0,16,56,8, ITEMTEXT |
 92                           HIGHCOMP | ITEMENABLED,
 93                           0,(APTR)&Clck3,NULL,0,NULL,0};
 94
 95 struct Menu ClckMenu = {NULL,491,0,48,8,MENUENABLED,
 96                        "Items",&Clck3It};
 97 /* Das sind Angaben fuer das gesamte Menue:                 */
 98 /* Zeiger auf naechstes Menue = Kein naechstes Menue.       */
 99 /* Rechte obere Ecke des Menues und die Ausdehnung          */
100 /* nach unten beim Anwaehlen. IDCMP Msg wird geschickt      */
101 /* wenn Menue gewaehlt wird. Name das gesamten Menues       */
102 /* und zum Schluss noch Zeiger auf das erste Item           */
103
104
105 main()
106 {
107 char *OpenLibrary();                      /*  Hier werden   */
108 struct MenuItem *item,*ItemAddress();     /*   Variablen    */
109 ULONG class, zmem, tt=0;                  /*   definiert    */
110 USHORT code, itemnum;                     /*                */
111 int mode=0;
112 long diff=0;
113
114 /*Jetzt werden die benoetigten Libraries und das Window     */
115 /*geoeffnet, anschliessend ein Timer Device aufgemacht      */
116 /*Bei einem Fehler schliesst das Prog. die bereits          */
117 /*geoeffneten Libraries sowie das Window und beendet        */
```

Listing 1. »Terminal-Clock« in C. Verwendet wurde der Aztek-Compiler.

```
118 /*das Programm ohne Meldung */
119
120 if(!(IntuitionBase = (struct IntuitionBase *)
121    OpenLibrary ("intuition.library", 1L)))
122   exit(FALSE);
123 if (!(GfxBase = (struct GfxBase *)
124    OpenLibrary ("graphics.library", 1L)))
125  {
126   CloseLibrary(IntuitionBase);
127   exit(FALSE);
128  }
129 if((mywind = (struct Window *)OpenWindow(&nw)) == NULL)
130  {
131   CloseLibrary(IntuitionBase);
132   CloseLibrary(GfxBase);
133   exit(FALSE);
134  }
135 if(!(timerport = CreatePort("Timer Port",0L)))
136  {
137   CloseWindow(mywind);
138   CloseLibrary(IntuitionBase);
139   CloseLibrary(GfxBase);
140   exit(FALSE);
141  }
142 if(OpenDevice(TIMERNAME, UNIT_VBLANK,
143               (char *) &timereq, 0L))
144  {
145   DeletePort(timerport);
146   CloseWindow(mywind);
147   CloseLibrary(IntuitionBase);
148   CloseLibrary(GfxBase);
149   exit(FALSE);
150  }
151 PrtTime(diff,mywind); /*Zeit ist Null und wird angezeigt*/
152 SetMenuStrip(mywind,&ClckMenu); /* Menue aktiv */
153 for(;;) /* Schleife laeuft bis Fenster geschlossen wird */
154 {
155  Wait((1L << mywind->UserPort->mp_SigBit) |
156       (1L << timerport->mp_SigBit));
157 /*Programm wartet auf Msg, Intuition und den Timer */
158
159   while(GetMsg(timerport))
160    if(mode)   /* wenn die Uhr lauft und Msg von Timer */
161     {
162      StartTimer();     /* Timer wird erneut gestartet */
163      diff=time(tt)-zmem; /* Zeit wird berechnet. */
164      if(diff>=360000 || diff<0)
165       {              /* wenn groesser 99h dann Null */
166        diff=0;
167        zmem=time(tt);
168       }
169      PrtTime(diff,mywind); /* Zeit ausgeben */
170     }
171   while(Msg=GetMsg(mywind->UserPort))
172     {     /* wenn Intuition Msg...*/
173   code=Msg->Code;
174   class=Msg->Class;
175   ReplyMsg(Msg);
176    if (class==MENUPICK)
177     { /* wenn Menue gewahlt...*/
178      if(code != MENUNULL)
179       { /* wenn Menuepunkt gewaehlt...*/
180        itemnum = ITEMNUM(code);
181        if(itemnum == 2)
182         {                          /* wenn Start gewaehlt...*/
183          Clck1It.Flags = Clck1It.Flags & ITEMTEXT |
184                                         HIGHCOMP;
185          /*...wird Start inaktiv und Stop aktiv */
186          Clck2It.Flags = Clck2It.Flags | ITEMENABLED;
187          StartTimer();
188          zmem=time(tt)-diff;
189          mode=1;
190         }
191        if(itemnum == 1)
192         {                          /* wenn Stop gewaehlt...*/
193          Clck1It.Flags = Clck1It.Flags | ITEMENABLED;
194          Clck2It.Flags = Clck2It.Flags & ITEMTEXT |
195                                           HIGHCOMP;
196          /*...wird Start aktiv und Stop desaktiviert */
197          mode=0;
198         }
199        if(itemnum == 0) /* Menue Null */
200         {                          /* wenn Null gewaehlt...      */
201          diff=0;
202          zmem=time(tt);
203          PrtTime(diff,mywind);
204         }
205       }
206     }
207    if (class==CLOSEWINDOW)
208     {
209      CloseDevice(&timereq);     /* Bei CloseWindow wird        */
210      DeletePort(timerport);     /* Speicher freigegeben        */
211      ClearMenuStrip(mywind);    /* und Programm beendet        */
212      CloseWindow(mywind);
213      CloseLibrary(GfxBase);
214      CloseLibrary(IntuitionBase);
215      exit(TRUE);
216     }
217    if (class==REFRESHWINDOW || class==ACTIVEWINDOW)
218     /* wenn Ziffern verdeckt werden wird neu geschrieben      */
219     PrtTime(diff,mywind);
220     } /*Menue !*/
221   } /*For Loop*/
222 } /*Main*/
223
224 PrtTime(diff,mywind)      /* Diese Routine gibt die Dif-       */
225                           /* ferenz der beiden Zeitvar.        */
226 long diff;                /* in h, min, s aus                  */
227 struct Window *mywind;
228 {
229 char s[8];
230 long h,m;
231 h=diff/3600;
232 diff-=h*3600;
233 m=diff/60;
234 diff-=m*60;
235 if(h<10)
236  {
237   s[0]='0';
238   s[1]='0'+h;
239  }
240 else
241  {
242   s[0]='0'+(h/10);
243   s[1]='0'+(h-((h/10)*10));
244  }
245 s[2]=':';
246 if(m<10)
247  {
248   s[3]='0';
249   s[4]='0'+m;
250  }
251 else
252  {
253   s[3]='0'+(m/10);
254   s[4]='0'+(m-((m/10)*10));
255  }
256 s[5]=':';
257 if(diff<10)
258  {
259   s[6]='0';
260   s[7]='0'+diff;
261  }
262 else
263  {
264   s[6]='0'+(diff/10);
265   s[7]='0'+(diff-((diff/10)*10));
266  }
267 s[8]='\0';
268 Move(mywind->RPort,30L,7L); /* Cursor wird positioniert,    */
269 SetAPen(mywind->RPort,1L);  /*    Schriftfarbe gewaehlt,    */
270 Text(mywind->RPort,&s,8L);  /*       Zeit geschrieben       */
271 }
272
273 StartTimer()  /* Hier wird der Timer auf etwas weniger      */
274 {             /* als eine Sekunde eingestellt und ge-       */
275               /* startet                                    */
276
277  timereq.tr_time.tv_secs = 0;
278  timereq.tr_time.tv_micro = 999500;
279  timereq.tr_node.io_Command = TR_ADDREQUEST;
280  timereq.tr_node.io_Flags = 0;
281  timereq.tr_node.io_Error = 0;
282  timereq.tr_node.io_Message.mn_ReplyPort = timerport;
283  SendIO((char *) &timereq.tr_node);
284 }

(c) M&T
```

Tips & Tricks zum Amiga

In einer eigenen Zeitschrift für alle Amiga-Fans darf natürlich auch eine Tips & Tricks-Rubrik nicht fehlen. Denn gerade die hier vorgestellten kurzen Hinweise erleichtern jedem Anwender den Umgang mit seinem Computer. Diesmal sind unter anderem einige grundlegende Beiträge zum CLI dabei.

Es sei an dieser Stelle nochmals darauf hingewiesen, daß Sie selbst durch Ihre Leserbriefe und Einsendungen den Inhalt dieser Rubrik mitgestalten können. Wenn Sie irgendeinen kurzen Tip oder auch ein Basic- oder sonstiges Listing ausgetüftelt haben, schreiben Sie uns. Jeder brauchbare Beitrag wird, natürlich gegen Honorar, in dieser Rubrik veröffentlicht.

Einsteiger-Tips zum CLI

Das erste, womit der Amiga-Anwender beim Laden einer Workbench- oder einer anderen, boot-fähigen Diskette konfrontiert wird, ist ein Fenster (Window) mit der Startmeldung. Erst später erscheint die Benutzeroberfläche mit den Diskettensymbolen. Die Workbench selbst ist nämlich auch nur ein Programm, das mit dem CLI-Befehl LOADWB geladen wird. Dieser Befehl steht in der »s/Startup-Sequence«-Datei, die sich auf jeder bootfähigen Diskette befindet. Drücken Sie beim Boot-Vorgang einmal die Tastenkombination <CTRL D>, sobald das Window auf dem Bildschirm erscheint. Es wird »***Break***« und das sogenannte Prompt ausgegeben. In unserem Fall ist das ein »1 > «. Wir befinden uns im Command Line Interface (CLI) des Amiga.

Das Prompt zeigt dem Anwender die Bereitschaft des Computers zu Tastatureingaben. Zusätzlich steht im Prompt vor dem Größerzeichen noch die Nummer des laufenden CLI-Task. Wir haben es bekanntlich mit einem Multitasking-System zu tun. Tippen Sie nun den Befehl NEWCLI ein — ein zweites Window erscheint, diesmal mit einem Prompt »2 > «. Sie können jetzt mit der Maus zwischen beiden CLIs wechseln (fahren Sie mit der Maus in das gewünschte CLI-Fenster und drücken Sie die linke Maus-Taste). Das ist sehr praktisch, wenn man beispielsweise mit dem DIR-Befehl das Disketteninhaltsverzeichnis auf das erste CLI-Window gelegt hat, während im zweiten CLI die nötigen Files mit »Copy« in eine RAM-Disk kopiert werden können.

Tippen Sie nun LOADWB ein. Damit haben Sie per CLI (das auch nur ein Teilprogramm des Amiga-DOS ist) das Workbench-Programm gestartet. Die Workbench wiederum ist ein Programm, das die Routinen der Benutzeroberfläche »Intuition« und der Systembibliothek »Icon« nutzt. Nach dem Aufruf der Workbench haben Sie (wenn Sie sich die Windows nach Wunsch entsprechend auf dem Bildschirm einrichten) nun die volle Leistungsfähigkeit eines Kommando-Interpreters und einer grafischen Benutzeroberfläche gleichzeitig. Diskettennamen ändern geht per Workbench schneller und komfortabler, wohingegen das CLI für verschiedene andere Arbeiten einfacher zu bedienen ist. Bleiben wir erst einmal beim CLI, doch ohne die Workbench zu vernachlässigen. Sehen wir uns einmal mit TYPE S/STARTUP-SEQUENCE an, was beim Booten passiert. Auf dem Bildschirm erscheint eine Reihe von Befehlen, unter anderem auch LOADWB und ENDCLI. Die Befehle, die in der Startup-Sequence stehen, werden beim Booten der Reihe nach ausgeführt. Natürlich können wir jederzeit die Befehlsfolge ändern, die beim Laden ausgeführt werden soll.

Mit der Befehlsfolge ED S/STARTUP-SEQUENCE rufen wir den Texteditor ED auf und finden in einem großen Window unsere Befehlsfolge wieder. Hier können wir nach Belieben Befehle ändern, hinzufügen oder entfernen. Entfernen wir doch einmal nur den ENDCLI-Befehl, indem wir den Cursor auf seine Zeile steuern und <CTRL B> (Zeile löschen) drücken. Mit der Escape-Taste (ESC) kommen wir in eine Befehlszeile, in der wir — nach Entfernung des Schreibschutzes von der Diskette — ein »X« eingeben und <RETURN> drücken. Die so geänderte Startup-Sequenz löscht uns bei einem Neubooten der Workbench nicht jedesmal den CLI, so daß wir immer gleich die Vorteile beider Benutzerschnittstellen gleichzeitig verwenden können. Ein Window mit Icons sagt uns nämlich nicht immer, welche Programme sich noch auf einer Diskette befinden. Oder hätten Sie als Einsteiger gewußt, daß es den Editor ED auf Ihrer Workbench-Diskette gibt?

Der erste Befehl, den man im Command-Line-Interface verwendet, ist in aller Regel der DIR-Befehl. Mit DIR OPT A kann man sogar das lästige einzelne Listen der Subdirectories vermeiden. Nur manchmal stört es ziemlich, wenn zusammengehörige Files ungeordnet — eines links, eines rechts — im Bild erscheinen. LIST QUICK schafft Abhilfe, alle Dateien erscheinen schön geordnet untereinander.

(Manfred Kohlen/tr)

Amiga als Multiuser-System

Auf einigen Disketten, zum Beispiel der Tool-Diskette von Metacomco, befindet sich im L-Ordner ein File names Aux-Handler. Mit dessen Hilfe können Sie Ihren Multitasking-fähigen Amiga sogar Multiuser-fähig machen! Alles, was Sie weiterhin dazu benötigen, ist ein zweiter Computer mit RS232-Schnittstelle und ein Nullmodem oder zwei Akustikkoppler. Verbinden Sie Ihren Amiga und den zweiten Computer über die RS232-Schnittstelle und geben folgendes ein (das File Aux-Handler muß sich, wie bereits gesagt, im L-Ordner der aktuellen Workbench-Diskette befinden):

```
mount aux:
newcli aux:
```

Wenn Sie alles richtig gemacht, und im Preferences-Menü die richtigen Übertragungsparameter eingestellt haben, sollte auf dem zweiten Computer das CLI-Prompt erscheinen. Jetzt können Sie sowohl von diesem wie auch vom Amiga aus zum Beispiel Diskettenoperationen ausführen. Achtung: Wenn Sie vom zweiten Computer aus zum Beispiel den Editor ED aufrufen, so erscheint dieser natürlich nur auf dem Amiga selbst. Lediglich alle CLI-Ein- und Ausgaben können Sie verwenden.

(Teut Weidemann/tr)

Schneller Subdirectory-Wechsel

Wenn Sie eine Diskette mit tief verschachtelten Subdirectories haben, kann der Wechsel von einem zum anderen Ordner zur langwierigen Prozedur werden. Aber anstatt zum Beispiel zweimal »cd /« einzugeben, genügt die Angabe von »cd //«. Angenommen, Sie befinden sich in einem Subdirectory der dritten Stufe und möchten in den C-Ordner: »cd ///c«. Natürlich funktioniert das auch beim DIR-Befehl: »dir ///c«.

(Teut Weidemann/tr)

Amiga als einfache Schreibmaschine

Wer nur eben mal einen kurzen Text oder Brief zu Papier bringen will, wird folgende CLI-Befehlssequenz zu schätzen wissen:

```
run type >prt: con:0/0/640/100/text->printer
```

Dadurch wird das CLI veranlaßt, alle Eingaben von der Tastatur (»con:« = »console« = Steuerpult, hier Tastatur) an den Drucker weiterzuleiten. Natürlich kann man damit eine schon gedruckte Zeile nicht mehr verbessern. Die Schriftart und die Druckparameter müssen vorher im Preferences-Menü eingestellt werden. Wer dies direkt am Drucker über Dip-Schalter tun kann, schickt den Text am besten gleich ohne Umwandlung durch den Amiga an die Druckerschnittstelle. Für Drucker, die an der Centronics-Schnittstelle angeschlossen sind, sieht die Zeile dann so aus:

```
run type >par: con:0/0/640/100/text->printer
```

Oder für Drucker am seriellen Port:

```
run type >ser: con:0/0/640/100/text->printer
```

(Reiner Mönig/tr)

Was tun bei »Task held«?

Wer einen schwerwiegenden Programmierfehler gemacht und den Amiga dadurch zum Absturz gebracht hat, kennt die Meldung »Task held, finish all Disc-activity«. Was viele aber nicht wissen: Diese Meldung besagt, daß lediglich dieser eine Task angehalten wurde. Wenn Sie zu diesem Zeitpunkt noch einen weiteren Task laufen haben, zum Beispiel ein zweites CLI, können Sie mit diesem problemlos weiterarbeiten. Vor allem Einsteiger schalten bei dieser Meldung nämlich den Computer aus und wieder ein, da sie meinen, er wäre vollständig abgestürzt. (Teut Weidemann/tr)

Tip bei LOAD-Errors

Insbesondere bei qualitativ schlechten Disketten kann das Amiga-Laufwerk schnell einen LOAD-Error melden. Manchmal läßt sich durch Anklicken der »RETRY«-Meldung das Programm nach einigen Versuchen doch noch laden. Einige Programme aber haben zum Beispiel die Bildschirmfarben so gewählt, daß Sie diese Meldung entweder nicht erkennen können, oder sie in einem verdeckten Window liegt. Die Tastenkombination <linke Amiga-Taste>-<V> (beim Amiga 500 und 2000 ist es die Taste mit dem Commodore-Zeichen) bewirkt dasselbe wie ein Anklicken der Retry-Meldung. (Teut Weidemann/tr)

Profi-Tips zum CLI

Haben Sie schon mal überlegt, warum Sie nach dem falschen Eintippen einer Zeile im CLI Ihre Backspace-Taste laufend so strapazieren? Wenn man schon früher gewußt hätte, daß <CTRL X> genügt, um die ganze Zeile zu löschen!

Und wenn Sie gleich eine ganze Reihe von — richtig eingetippten — Befehlen ausführen lassen wollen, brauchen Sie nicht unbedingt alles einzeln eintippen und auf die Bearbeitung jedes Befehls zu warten. Für eine automatische Abarbeitung der CLI-Befehle gibt es nämlich drei Möglichkeiten — eine davon findet man in keinem Handbuch, nicht einmal in der Entwickler-Dokumentation.

Möglichkeit 1: Häufig benutzte Befehlsfolgen legt man mit dem Editor ED in einem File ab, welches mit dem EXECUTE-Befehl aufgerufen werden kann und dann selbständig seine Arbeit verrichtet. Bei sehr häufiger Verwendung des EXECUTE-Befehls empfiehlt sich, den Befehlsnamen im C-Directory per RENAME zu kürzen. Profis rufen eine Datei zum Kopieren der DOS-Befehle ins RAM nur noch mit »X Y« auf (X für EXECUTE und Y als Filename). Die Amiga-DOS-Datei Y sollte dann alle Befehle zum Kopieren der wichtigsten Befehle ins RAM enthalten.

Möglichkeit 2: Die zweite Methode, Befehlsfolgen aufzurufen, findet sich im englischsprachigen DOS-Manual oder dem deutschen Amiga-Handbuch (Markt&Technik Verlag): Mit RUN wird ein Task gestartet, der die Befehle ausführt, die man per Pluszeichen und RETURN voneinander trennt. Das sieht dann so aus:

```
RUN COPY Y TO PRT+
LIST+
NEWCLI
```

Das CLI überprüft die Eingabe auf das Pluszeichen und startet den Task erst, wenn ein Befehl ohne Pluszeichen eingegeben wird.

Möglichkeit 3: Jedesmal das RUN und die Pluszeichen einzugeben, ärgert sehr, wenn man nicht unbedingt einen Task dafür braucht. Abhilfe schafft die Tastenkombination <CTRL J>. Tippen Sie einfach den ersten Befehl der Folge (ohne RUN), drücken dann aber anstatt der RETURN-Taste <CTRL J>. Der Cursor ist in der nächsten Zeile, der Befehl wird aber noch nicht ausgeführt. Jetzt den nächsten Befehl eingeben, besagte Kontrollfunktion, und so weiter, bis zum letzten Befehl. Erst jetzt <RETURN> drücken. Probieren Sie es aus!

Nun stellen Sie sich einmal vor, in der obersten Zeile des CLI-Windows steht ein wichtiger Text, zum Beispiel ein jetzt benötigter Dateiname oder ähnliches. Sie müssen aber unbedingt noch etwas eintippen, sind dummerweise in der letzten Zeile des CLI-Windows angekommen und würden den obenstehenden Text bei einer weiteren Befehlseingabe verlieren. Wenn Sie Glück haben und Ihr CLI-Window noch nicht die volle Größe hat, können Sie dadurch ein Wegscrollen des Textes verhindern. Ansonsten gibt es noch die Tastenkombination <CTRL K>: Der Cursor wandert eine Zeile nach oben. Auch wenn in der jetzt neuen Eingabezeile schon etwas stehen sollte, wird vom Amiga-DOS nur das übernommen, was Sie jetzt eingeben. Natürlich können Sie damit nicht zum Beispiel ein paar Zeilen höher gehen, um dort eine falsche Eingabe zu verbessern. Das CLI ist und bleibt ein Zeileneditor. Mit <CTRL J> wandert der Cursor übrigens genauso wieder nach unten, wie er das aufwärts auch tat — ohne Texte mit neuen darübergetippten Befehlen unkenntlich zu machen.

Am Ende solcher Kontrolloperationen und vor einer neuen Befehlseingabe empfiehlt sich übrigens dringend die Tastenkombination <CTRL X>. Manchmal versteht das CLI verschiedene Kombinationen mit der CTRL-Taste nämlich als Buchstaben, und einen Befehl namens »<CTRL G>-DIR« gibt es bekanntlich nicht.

Ihnen ist sicher schon oft aufgefallen, daß beim Aufrufen der Befehle immer auf Diskette zugegriffen wird. Das CLI ist nämlich ein kurzes Programm, das neben ein paar Kontroll- und Überwachungsfunktionen nichts weiter tut, als Programme nachzuladen und auszuführen. Die Befehle des Amiga-DOS sind eben auch nur Programme. Deswegen kann man auch jeden Programmnamen eintippen, ohne extra anzugeben, daß man dieses Programm jetzt starten möchte. Wer früher auf einem C 64 oder ähnlichen Heimcomputern gearbeitet hat, wird also auf den gewohnten LOAD-Befehl verzichten müssen.

Findet das CLI das aufgerufene File nicht, sieht er zuerst im C-Directory der Systemdiskette nach — dort stehen im Normallfall die Befehle. Erst nach einem weiteren Mißerfolg wird eine Fehlermeldung ausgegeben.

Das ständige Nachladen strapaziert Nerven, Diskette und Laufwerk gleichermaßen. Viel einfacher geht es, wenn wir diese Programme in die RAM-Disk kopieren. »MAKEDIR RAM:C« eröffnet ein Directory »C« in der RAM-Disk, »COPY C: TO RAM:C« kopiert alle CLI-Befehle ins RAM. Doch das CLI denkt noch immer nicht daran, die Befehle nun aus der RAM-Disk zu holen. Wir müssen noch den Zugriff des CLI auf das C-Directory der RAM-Disk verlegen: »ASSIGN C: RAM:C«. Zugegeben, diese Prozedur benötigt viel Speicher. Man sollte also möglichst nur die wichtigen, ständig benötigten Befehle in die RAM-Disk kopieren. Wenn Sie etwas Erfahrung im Umgang mit dem CLI haben, kennen Sie selbst die Befehle, die Sie am häufigsten benötigen. Man legt dazu einfach die entsprechende Folge von COPY-Befehlen in einem File ab, das jederzeit mit EXECUTE ausgeführt werden kann. Zur Zeitersparnis sollte man also zuerst die Befehle COPY und ASSIGN ins RAM kopieren und dann bereits das C-Directory auf das RAM umstellen. Die restlichen Copy-Befehle werden jetzt schon wesentlich schneller ausgeführt, da sie bereits vom RAM ausgehen. Übrigens: Wenn Sie nur ASSIGN eingeben, sehen Sie, auf welches Directory in welchem Laufwerk das CLI nun eingestellt ist.

(Manfred Kohlen/tr)

Der Profi-Tip: Eingebauter Debugger beim Amiga

Wußten Sie schon, daß im Amiga-Betriebssystem (im Kickstart-ROM) ein Debugger eingebaut ist? Er heißt »ROM-Wack« und ist ein Subset des Debuggers, der auf der Makroassembler-Disk aus dem Entwicklerpaket unter dem Namen WACK gespeichert ist. ROM-Wack wird bei einem »Guru Meditation Error« angesprungen, wenn am RS232-Port des Amiga ein Terminal angeschlossen ist (vielleicht sogar ein alter C 64 mit Terminalprogramm oder ein unwürdiger MS-DOS-Clone). Über dieses Terminal kann dann im Speicher des Amiga auf Fehlersuche gegangen werden.

Was die wenigsten wissen: ROM-Wack läßt sich auch vom CLI aus aufrufen. Ist die Workbench noch nicht geladen (also <CTRL D> beim Booten), kann mit dem Befehl »LOADWB -DEBUG« das ROM-Wack als Bestandteil der Workbench geladen werden. Probieren Sie jetzt bei gedrücktem rechten Mausknopf auf der Workbench-Menüzeile entlang nach rechts zu fahren. Ganz rechts erscheint plötzlich ein vorher verstecktes Menü. Damit können Sie das über die serielle Schnittstelle laufende ROM-Wack aktivieren und steuern, auch wenn der Amiga bereits keinen Ton mehr von sich gibt (sprich: rettungslos abgestürzt ist). Eine ausführliche Anleitung zum ROM-Wack findet sich im (englischsprachigen) »Amiga-DOS Developers Manual«. (Manfred Kohlen/tr)

Schnelle Compiler

AMIGA TEST Sieht man schnelle Grafik auf dem Amiga, denkt man unwillkürlich, das Programm sei in Assembler geschrieben. Aber auch mit der Programmiersprache »C« sind sehr flotte Programme möglich.

Das ABasic auf dem Amiga ist zwar eine Programmiersprache, die viel Komfort bietet, aber für Anwendungen mit Grafik ist sie einfach zu langsam. Außerdem besteht nur ein eingeschränkter Zugriff auf die Fähigkeiten des Amiga-Betriebssystems. Will man trotzdem nicht in Assembler programmieren, bietet sich eine andere Hochsprache an : »C«. Sie verbindet hohe Geschwindigkeit mit maschinennaher Programmierung und ist auf dem Amiga mit dem in C geschriebenen Betriebssystem gut geeignet, professionelle Programme zu schreiben.

Die zwei getesteten Compiler waren der Lattice Version 3.10 und der Manx Aztec Version 3.40A, der seit kurzem ausgeliefert wird. Beide Compiler halten sich an den Standard von Kerningham und Ritchie, der Lattice-C-Compiler beinhaltet zusätzlich noch viele der heute in Unix-Systemen verwendeten Befehle.

Beide Pakete arbeiten unter dem CLI, da die Bedienung der Programme mit der Workbench gar nicht zu verwirklichen ist. Dennoch können die Vorteile des Amiga genutzt werden. Zum Beispiel lassen sich mehrere Fenster für verschiedene Zwecke aufmachen. Es spart viel Zeit, wenn in einem Fenster die Textverarbeitung mit dem aktuellen Sourcecode steht, während in einem anderen Compiler und Linker aufgerufen werden können. Man spart sich also bei erneuten Verbesserungen des Quelltextes die Ladezeit für Editor und Sourcecode.

Wenn man einen Compiler kauft, sollte man auch darauf achten, was mitgeliefert wird. Denn manchmal ermöglichen die Programme, die sich zusätzlich auf der Diskette befinden, ein leichteres und schnelleres Arbeiten. Bei beiden Compilern stehen im Subdirectory »c« die gängigen Amiga-DOS-Befehle, beim Aztec-Compiler sind einige mehr vorhanden. Auf der Diskette von Lattice stehen in diesem Directory auch der Compiler, der Linker und der Disassembler. Diese drei Programme befinden sich beim Aztec zusammen mit dem Assembler in einem eigenen Subdirectory namens »bin«. Das hat den Vorteil, daß zum Beispiel ein ASSIGN-Befehl nur auf die obengenannten Teile des Compilers wirkt und nicht auf alle Befehle. Dadurch ist man flexibler in der Aufteilung der Directories. So kann das Subdirectory in die RAM-Disk kopiert werden. Danach teilt man dem Betriebssystem die neue Lage mit »assign bin: ram:bin« mit. Der Compiler und der Linker werden jetzt aus der RAM-Disk geladen, die Amiga-DOS-Befehle werden jedoch nach wie vor im Directory »c« gesucht.

Beide Pakete beinhalten die notwendigen Amiga-spezifischen Includefiles von Commodore. Die bei allen C-Compilern standardmäßig vorhandenen Headerfiles fehlen natürlich auch nicht. Die Struktur und der Inhalt dieser Dateien ist bei beiden Programmpaketen gleich.

Wichtig: Bibliotheken

Die Bibliotheken enthalten Funktionen, die speziell für den Amiga geschrieben wurden. Hier findet man zum Beispiel die Mathematikroutinen zur Berechnung von Sinus und Cosinus. Alle diese Funktionen sind in Assembler geschrieben, um eine hohe Geschwindigkeit zu erreichen. Bei diesen Bibliotheken werden die ersten Unterschiede zwischen den zwei Produkten sichtbar. Der Lattice-Compiler besitzt zwei verschiedene Bibliotheken, die für die Fließkomma-Arithmetik einsetzbar sind. Die erste, »lcmffp.lib«, unterstützt einen Teil der Fast-Floating-Point-Routinen, die von Commodore mitgeliefert werden. Die andere Bibliothek (»lcm.lib«) enthält alle Mathematikroutinen, die im Handbuch beschrieben sind, und arbeitet im IEEE-Format.

Diese zwei wichtigen Teile sind auch beim Aztec-Compiler in angepaßter Version zu finden, aber die Namen sind verändert (»m.lib«,»ma.lib«). Zusätzlich gibt es noch eine Version der IEEE-Bibliothek (»mx.lib«), die speziell für den Aztec-Compiler geschrieben wurde und deshalb schneller ist als die normale IEEE-Bibliothek. Allerdings haben die Entwickler streng zwischen den verschiedenen Speichermodellen unterschieden, was das Verständnis erleichtert. Bei den zwei Modellen können Daten und Programme entweder so organisiert sein, daß sie beliebig oder maximal 64 KByte groß sein dürfen. Die Wahl des Modells hat Auswirkungen auf die Länge und die Geschwindigkeit des fertigen Programms. Während die Adressen bei dem Modell mit beliebiger Größe 4 Byte umfassen, sind sie im kleineren nur 2 Byte lang. Als Voreinstellung werden für Daten sowie Programm die kleinen Modelle verwendet. Eine weitere Bibliothek existiert für den Gebrauch von Integer-Variablen, die 32 Bit lang sind. Für diese unterschiedlichen Modelle gibt es auf der zweiten Diskette die angepaßten Bibliotheken.

Interessant für viele Anwender dürfte auch die Bibliothek »s.lib« für den Aztec-Compiler sein, die 15 Funktionen für die Bildschirmausgabe enthält. Vom Bildschirmlöschen bis zum Setzen des Cursors auf eine bestimmte Position, ist alles vorhanden. Gerade für Anfänger ist diese Sammlung von Funktionen sehr nützlich, da man sich dadurch sehr viel Arbeit spart.

Bei Aztec muß man zwischen den zwei lieferbaren Versionen unterscheiden. Die kleine Version enthält bereits alle nötigen Programme. Auf der zweiten Diskette befinden

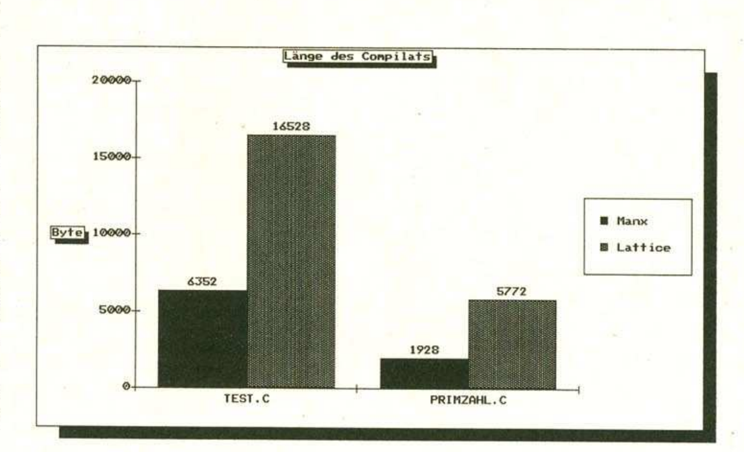


Bild 1. Deutliche Unterschiede sind bei der Länge der fertigen Programme zu sehen

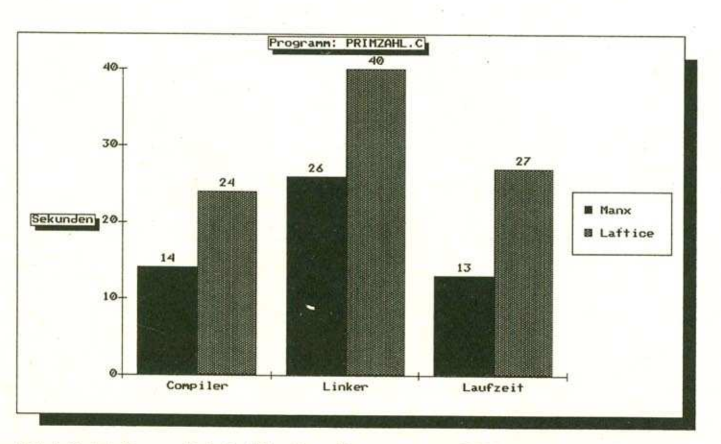


Bild 2. Zeitvergleich für das Programm PRIMZAHL.C

sich einige nützliche Programme und die Sourcecodes für viele Assembler-Funktionen. Bei der größeren Version findet man zusätzlich die Sourcecodes zu den Bibliotheken, einen Texteditor und mehrere Utilities, die von Unix übernommen wurden und die Arbeit erleichtern. Bei Lattice sind zwar auch einige Produkte (Editor, verbesserter Linker etc.) vorhanden, die zusätzlich gekauft werden können, aber am Anfang wirkt das gelieferte Paket eher spartanisch.

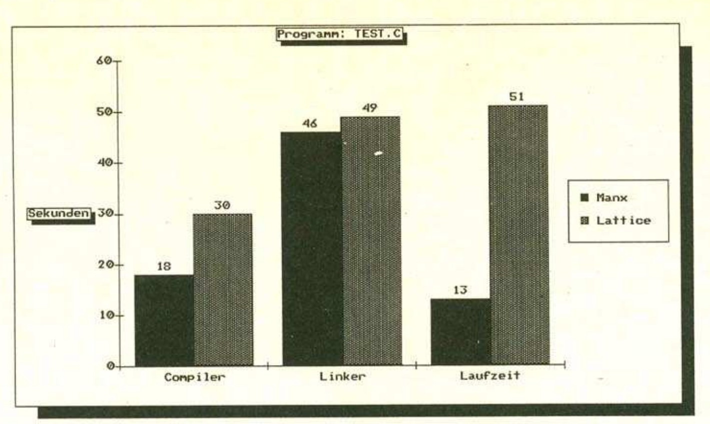


Bild 3. Die benötigten Zeiten zum Compilieren, Linken und Abarbeiten des Programms TEST.C

Hilfsprogramme

Eines der Utilities, die viel Arbeit und Zeit sparen können und beim Aztec-C mitgeliefert werden, ist »MAKE«. Normalerweise werden große Programme in mehrere kleine Teile zerlegt und getrennt compiliert. Wenn man nun eine Änderung an einem der Source-Texte vornimmt, muß eigentlich nur dieser compiliert und zu den unveränderten Objektdateien gelinkt werden. Mit einer Batchdatei (Datei mit Befehlen) ist dies nicht zu verwirklichen, hier werden alle Dateien neu compiliert, was unnötig Zeit kostet. An dieser Stelle lohnt sich der Einsatz von MAKE. Zuerst sucht das Programm eine Datei namens »MAKEFILE«, in die der Programmierer vorher schon eintragen muß, welche Datei auf welcher aufbaut und welche Dateien zu erzeugen sind. MAKE compiliert jetzt nur alle Dateien, bei denen der Sourcecode später auf Diskette geschrieben wurde, als das dazugehörende Objekt-File. Voraussetzung für die richtige Funktion ist allerdings, daß das Datum und die Uhrzeit immer richtig gesetzt sind, MAKE entscheidet nämlich aufgrund des eingetragenen Datums.

Für die Environment-Variablen (Environment = Umgebung), gibt es bei Manx noch ein besonderes Programm. Mit »SET« kann man diese Variablen ändern, die bestimmen, wo der Compiler seine Include-Dateien sucht, wo die Zwischendateien angelegt werden sollen und vieles mehr. Durch Aufruf von SET ohne Parameter werden alle Environment-Variablen und ihr Wert ausgegeben.

Mit einem weiteren Utility, dem symbolischen Debugger, ist es sogar möglich, »abgestürzte« Programme nachträglich zu überprüfen. Eine Voraussetzung ist allerdings, daß man den Debugger noch aufrufen kann, was ein CLI-Fenster voraussetzt, von dem aus noch kein Task gestartet wurde. Dadurch kann unter Umständen sogar der Fehler für den Absturz gefunden werden. Dies funktioniert allerdings nur, wenn durch den Fehler nicht der Speicher durcheinandergebracht wurde. Sogar an die Verwendung des Mathematik-Coprozessors MC68881 ist gedacht worden. Benutzt das Programm den Coprozessor, erfolgt auch die Anzeige seiner Register.

Noch ein Tip, da es gerade um Abstürze geht: Viele C-Programme stürzen ab, weil die Return-Codes von angesprungenen Routinen nicht ausgewertet werden. Egal, ob Sie ein Fenster mit »OpenWindow« öffnen oder Speicher mit »AllocMem« belegen wollen, das Betriebssystem liefert eine »NULL« (ein definierter Wert, der bei den meisten C-Compilern eine 0 ist), wenn etwas schiefgelaufen ist. Werten Sie den Return-Code aus und brechen Sie ihr Programm ab, wenn Sie eine NULL zurückbekommen. Es kann nämlich zum Systemabsturz kommen, wenn Sie in ein Fenster schreiben, das gar nicht existiert. Und vergessen Sie nie, belegten Speicher mit »FreeMem« freizugeben und offene Bibliotheken zu schließen. Sie vermeiden dadurch, daß nicht mehr benutzbare Fenster existieren. Vom Programm belegte und nicht mehr freigegebene Speicherbereiche können nach Beenden des Programms nur noch nach einem Reset verwendet werden.

Übersetzung des Programms

Angenommen, man hat den Sourcecode schon eingegeben, so ist der nächste Schritt zum fertigen Programm das Compilieren. Die vielen Optionen, die bei beiden Compilern zur Verfügung stehen, unterscheiden sich nur geringfügig. Bei dem Compiler von Lattice muß allerdings die Reihenfolge von Flags und Dateinamen streng eingehalten werden. Beim Aztec-Compiler kann man außerdem vorcompilierte Dateien erzeugen und später einfach einbinden. Zum Beispiel compiliert man eine Datei, in der alle nötigen INCLUDE-Anweisungen für das Programm stehen. Wird diese Datei dann später mit einer Compileroption übernommen, werden schon bearbeitete INCLUDE-Befehle einfach überlesen, was natürlich enorm Zeit sparen kann. Ein wichtiges Merkmal des Aztec-Compilers ist noch zu erwähnen, die Um-

AMIGA TEST

schaltung von 16 Bit auf 32 Bit für die Darstellung von Integerzahlen. Als Voreinstellung wird die 2-Byte-Darstellung verwendet, was allerdings beim Compilieren von Programmen, die für den Lattice-Compiler geschrieben wurden, Probleme gibt. Man sollte also, wenn nicht speziell für den Aztec-Compiler gearbeitet wird, immer mit der »+l« Einstellung compilieren.

Beim Amiga gibt es ja zwei unterschiedliche Speicher, das Chip- und das Fast-Memory. Im Fast-Speicher laufen die Programme schneller, da nur der Prozessor darauf zugreifen kann. Aber manche Daten müssen im Chip-Memory liegen, da die Custom-Chips nur diese 512 KByte adressieren können. Beim Lattice muß die Festlegung, welche Daten wo stehen sollen, schon beim Compilieren getroffen werden, beim Aztec gibt man die entsprechenden Flags erst beim Linker an.

Doch jetzt zum Compilier-Vorgang selbst, wobei nur der Quellcode, der Objektcode und das fertige Programm in die RAM-Floppy geschrieben wurden. Es ergab sich, daß der Aztec-Compiler dem von Lattice deutlich an Geschwindigkeit überlegen ist. Bei sehr kleinen Programmen kann es durchaus passieren, daß der Lattice die doppelte Zeit beansprucht, bis das linkbare File vorliegt. Dieser Vorsprung wird zwar prozentual bei großen Programmen geringer, aber mit der schon angesprochenen Fähigkeit des Aztec-C-Compilers, Dateien vorzucompilieren, kann man auch noch Zeit sparen.

Sehr viel kürzere Zeiten für das Compilieren und Linken erreicht man bei beiden Paketen, wenn sowohl Compiler wie auch Linker in der RAM-Disk stehen, was allerdings nur mit mindestens 1 MByte Speicher möglich ist. Übersetzungszeiten für kurze Programme liegen dann im Bereich unter zehn Sekunden. Bei längeren Programmen können aber Probleme mit dem Speicherplatz auftreten.

Eine Möglichkeit, die besonders bei zeitkritischen Programmen interessant ist, wird von beiden Compilern geboten, und zwar Assembler-Code direkt in das C-Programm zu schreiben. Der Aztec-Compiler bietet mit einem weiteren Flag (»-a«) die Möglichkeit, den Assembler nicht zu starten. Dadurch wird nur der Assembler-Quellcode auf der Diskette erzeugt, der dann noch verändert werden kann.

```
#include <stdio.h>
#include <math.h>

#define MAX 32000

   char     feld[MAX];

main()
{
   int      i,l;
   double   z;

   printf("\n INTEGER");
   for(i=0;i<10000;i++)
   {
      l=3*i;
   }

   printf("\n DOUBLE");
   for(i=0;i<10000;i++)
   {
      z=sin((double)(i))*3;
   }

   printf("\n FELD");
   for(i=0;i<=MAX;feld[i++]=MAX-i);

   printf("\n FERTIG!");
}
```

Listing 1. Ein kleines Testprogramm, das mit Real- und Integerzahlen und einem Array arbeitet

```
main()
{
    char      feld[3500];
    char      *ptr;
    int       zahl,teil;

    ptr=feld;
    for (zahl=5;zahl<32000;zahl+=2)
    {
        teil=3;
        while (teil*teil<=zahl && zahl%teil!=0)
            teil+=2;
        if (teil*teil>zahl)
            *ptr++=zahl;
    }
}
```

Listing 2. Programm zur Berechnung der Primzahlen bis 32000

Zusammenfügen der Einzelteile

Liegen nun alle Programmteile als Objektmodule vor, müssen sie noch mit den benötigten Bibliotheken zusammengelinkt werden, um ein ablauffähiges Programm zu erhalten. Auch hier ist der Aztec wieder schneller, allerdings sind die Zeitunterschiede nicht so groß wie bei den Compilern. Was beim Linker von Lattice unangenehm auffällt, ist, daß man alle Anweisungen ausschreiben muß. Dadurch wird der Aufruf des Linkers sehr lang, was bei Eingabefehlern sehr ärgerlich ist, da man alles nochmal eingeben muß. Anders ist die Methode, die bei dem Aztec-Linker verwendet wird, sie erinnert stark an Unix. Als Kennzeichnung für Flags wird entweder <-> oder <+> verwendet, danach kommt das Flag (ein Buchstabe) und dann eventuell nötige Parameter. Die normale Befehlszeile beim Lattice-Linker sieht ungefähr so aus:

blink c.o,test.o lib:lc.lib,lib:amiga.lib

Ein vergleichbarer Aufruf des Aztec-Linkers ist da schon wesentlich kürzer:

ln test.o -lc

Compiler und Linker wurden seit der Version 3.20A bei Aztec noch erheblich verbessert. Unter anderem können mit der neuen Version jetzt auch beliebige Treiber (Druckertreiber etc.) gelinkt werden. Bei beiden Linkern müssen jedoch unbedingt die Fließkomma-Bibliotheken als erste angegeben werden.

Nun zu einem Punkt, der jeden sicher interessiert, die Größe des fertigen Programms. Die Unterschiede sind hier zum Teil ganz erheblich. Alles in allem erzeugt Aztec-C einen wesentlich kompakteren Code, was sich in kürzeren Ladezeiten bemerkbar macht. Aber auch der geringere Speicherplatzbedarf (siehe Bild 1) sollte nicht unterschätzt werden.

Große Überraschungen kann man auch mit der Geschwindigkeit erleben. Die Programme, die der Compiler von Lattice erzeugt, benötigen zur Abarbeitung teilweise viermal so viel Zeit, wie die des Aztec-Compilers (siehe Bild 2 und 3). Zwei der verwendeten Testprogramme finden Sie in Listing 1 und 2.

Ein Punkt, der sehr wichtig ist, kam bis jetzt noch nicht zur Sprache, die nötige Hardware für die Compiler. Mit dem Paket von Aztec ist es möglich, auch mit nur einem Laufwerk C-Programme zu schreiben. Die Vorgehensweise für diesen Fall ist im Handbuch beschrieben, sinnvoll ist jedoch ein zweites Laufwerk und 1 MByte Speicher, da dann alles etwas schneller geht. Mit dieser Ausstattung kann man auch sehr große Programme compilieren und linken. Etwas anders liegt die Sache bei dem Lattice-Compiler. Ohne zweites Laufwerk muß man schon etwas zaubern, um alles zum Laufen zu bringen. Bei wirklich großen Programmen bleibt der Kauf einer Festplatte unumgänglich.

Für den Heimanwender dürfte der Aztec-Compiler die bessere Lösung sein, obwohl das Grundpaket teurer ist, als das von Lattice. Allerdings liefert Lattice einen größeren Sprachumfang, der größtenteils von Unix übernommen wurde. Ein Beispiel dafür ist das Erzeugen eines zweiten Programms, das unabhängig vom ersten läuft. Man kann beispielsweise eine Diskette formatieren lassen, während das eigentliche Programm weiterläuft.

Sowohl Aztec-C als auch Lattice-C sind für professionelle Software-Entwicklung geeignet. Der modernere Sprachumfang wird dabei von Lattice-C geboten, wobei er auch billiger ist (Lattice: 498 Mark, Aztec klein: 690 Mark, groß: 1190 Mark). Aztec-C ist im Gegensatz dazu kleiner, schneller und erzeugt einen kompakteren Code, der mit kürzeren Laufzeiten glänzt. Bei all diesen Unterschieden bleibt er jedoch weitestgehend kompatibel zum Lattice-C. Welcher der beiden Compiler nun besser ist, läßt sich nicht ohne weiteres sagen, es hängt zu stark davon ab, was der Anwender wünscht. Jedenfalls laufen sie auf dem Amiga 1000 (Kickstart 1.2 und Workbench 1.2) sowie auf dem Amiga 500 einwandfrei. Der Zugriff auf die herausragenden Fähigkeiten des Amiga ist jedenfalls mit der Programmiersprache C leicht zu bewerkstelligen, wobei nur noch Assembler schneller ist.

(rb)

Anbieter:
Softwareland, Schweiz, CH-8050 Zürich, Franklinstraße 27, Tel. 0041-1-3115959
Philgerma GmbH & Co.KG, 8000 München 40, Ungererstraße 42, Tel. 089/395551
Interplan GmbH, 8000 München 19, Nymphenburgerstraße 134, Tel. 089/1234066

CAD auf dem Amiga

AMIGA TEST **CAD — dahinter verbirgt sich der Begriff computerunterstütztes Zeichnen und Konstruieren. Wir stellen Ihnen hier ein Programm vor, mit dem Sie ohne großen Aufwand Schaltpläne oder Konstruktionszeichnungen erstellen können.**

Der Computer ersetzt immer mehr Aufgaben, die noch vor nicht allzu langer Zeit mit anderen Mitteln erledigt wurden. So auch im CAD-Bereich. Der Begriff CAD steht für »Computer Aided Design« und meint nichts anderes als das Zeichnen und Konstruieren mit dem Computer. Zeichnete man bis vor ein paar Jahren noch diverse Pläne mit Tusche auf Papier, so wird in der heutigen Zeit mehr auf dem Bildschirm gearbeitet. Auch für den Amiga ist ein CAD-Programm erhältlich, mit dem sich solche Aufgaben erledigen lassen: »Aegis Draw Plus«.

Das Programm dient in erster Linie dazu, Schaltpläne, Grundrisse von Gebäuden oder Konstruktionszeichnungen zu erstellen. Dazu bietet Draw Plus eine Vielzahl von Hilfsfunktionen, die den Anwender unterstützen.

Über einfache Befehle lassen sich Linien ziehen, Kreise setzen oder Rechtecke in das Arbeitsfenster »montieren«. Dabei kann mit verschiedenen Linienbreiten, diversen Füllmustern (für Schraffuren), unterschiedlichen Stricharten (gepunktet, durchgehend oder gemischt) oder frei wählbaren Farben gearbeitet werden. Auch Texte lassen sich einsetzen. Da wegen der begrenzten Auflösung des Bildschirms leider nicht alle Details klar zu erkennen sind, lassen sich Teilbereiche der zu bearbeitenden Fläche auch vergrößern (Zoom). Dabei gehen keine Details verloren.

Draw Plus versucht den Nachteil der geringen Auflösung zwar dadurch zu vermeiden, daß das Programm im Interlace-Modus arbeitet, kann damit aber das Manko nicht beseitigen. Der Interlace-Modus wirkt sich sogar eher störend aus, da das Bild teilweise sehr flimmert. Man kann aber auch auf eine flimmerfreie Darstellung umschalten, bei der dann allerdings Details nicht mehr so gut wiedergegeben werden.

Aegis Draw Plus legt die Daten aller Linien und Zeichen als Vektoren ab. Dies bringt den Vorteil, daß alle Vergrößerungen oder Verkleinerungen ohne Qualitätsverluste durchgeführt werden, da nur die Verhältnisse der Zahlen zueinander umgerechnet werden müssen. Leider verlangsamt diese Art der Datenhandhabung die Arbeitsgeschwindigkeit des Programms, da das Programm vor dem eigentlichen Zeichnen erst umfangreiche Tabellen erstellen muß. So benötigt ein Bild, das von Diskette geladen wird, geraume Zeit, bis es nach Abschluß des Ladevorgangs auf dem Bildschirm erscheint. Draw Plus stellt aber auch die Möglichkeit zur Verfügung, die Bilder im gängigen IFF-Format zu speichern. Diese Bilder können damit beispielsweise auch von Deluxe-Paint, Graphicraft oder Aegis Images genutzt werden.

Soll also beispielsweise eine große Maßstabszeichnung angefertigt werden, bei der irgendwo Angaben zur Zeichnung stehen sollen, vergrößert man sich den entsprechenden Bereich und schreibt die betreffenden Daten hinein. Im unvergrößerten Gesamtbild wäre dann an dieser Stelle nur ein kleiner Fleck zu sehen, der aber alle Daten enthält.

Selbstverständlich können Linien oder Zeichen auch wieder aus der Zeichnung entfernt werden. Dazu ist nur der betreffende Bereich zu markieren.

Diverse Hilfsfunktionen wie etwa spiegeln oder drehen unterstützen den Anwender auch bei schwierigen Operationen.

Das Programm eignet sich übrigens nicht in Zusammenarbeit mit Druckern. Es läßt sich zwar die Hardcopy-Routine der Workbench ansprechen, aber das Resultat ist nur eine Kopie des Bildschirms. Details gehen durch die geringe Auflösung eines Druckers fast gänzlich verloren.

Draw Plus ist vielmehr auf den Umgang mit Plottern ausgelegt. Dies äußert sich schon darin, daß fast alle gebräuchlichen Plottertreiber mitgeliefert werden. Außerdem hätte es auch wenig Sinn, maßstäbliche Zeichnungen oder Konstruktionsunterlagen auf einem Drucker auszugeben. Zusätzlich besteht beim Arbeiten mit einem Plotter die Möglichkeit, die Abmessungen des Ausdrucks und andere, wichtige Details nach eigenen Vorstellungen einstellen zu können. Sie können also über einen kleinen Befehl angeben, in welchem Format das Bild ausgegeben werden soll.

Beim Kauf von Aegis Draw Plus dürfte also das Vorhandensein eines Plotters Voraussetzung sein. Die Bilder 1 und 2 zeigen Ihnen, wie gut solche geplotteten Zeichnungen aussehen können.

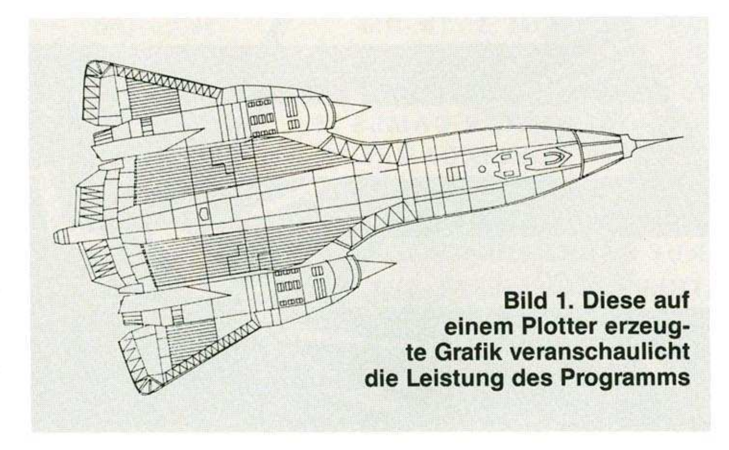


Bild 1. Diese auf einem Plotter erzeugte Grafik veranschaulicht die Leistung des Programms

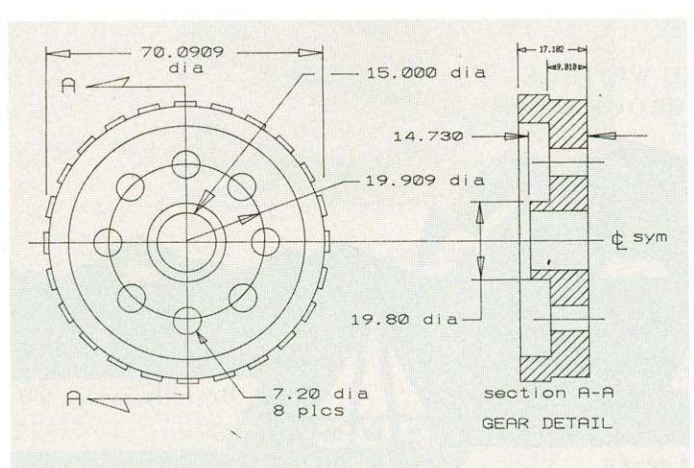


Bild 2. Konstruktionszeichnungen mit Bemaßungen sind mit Aegis Draw Plus kein Problem

Fazit

Die Software an sich zeigt sich durchdacht und bietet dem CAD-Anwender vielfältige Möglichkeiten. Eine genaue Besprechung aller zur Verfügung stehenden Punkte würden den Rahmen dieses Testberichts allerdings sprengen.

Leider ist es im Sinne eines brauchbaren Ausdrucks nötig, sich einen Plotter anzuschaffen. Die Druckerroutine hilft hier nicht weiter. Es ist zwar prinzipiell möglich, Drucker anzusteuern, doch dürfte das Resultat keineswegs den Anforderungen entsprechen.

Der Preis von etwas über 500 Mark mag dem einen oder anderen vielleicht etwas hoch erscheinen. Bedenkt man aber die Möglichkeiten (Arbeitserleichterung, präzise Ausdrucke, genaue Konstruktionen), kann Aegis Draw Plus diesen Preis durchaus rechtfertigen. (dm)

Bezugsquellen: Alle gutsortierten Softwareanbieter. Einzelne Adressen entnehmen Sie bitte dem Anzeigenteil.

AMIGA-WARE

A67 AEGIS DRAW PLUS 560.-DM
A03 AEGIS ANIMATOR MIT IMAGES 265.-

S64 SDI 95.-DM
S27 SEVEN CITIES OF GOLD 80.-DM

A65 AMIGADOS EXPRESS 60.-
A41 DELUX PAINT II 265.-DM

A00 SUPERBASE (deutsch) 249.-
A99 UPM 2.2 TEXT 249.-DM

SILICON DREAMS 65,– DM
S59 SINDBAD 95.-DM

S71 UNINVITED 110.-DM
S66 KINGS QUEST II 95.-DM

S57 SPACE QUEST 115.-DM
S29 STARFLEET I 120.-DM

S56 KINGS QUEST III 95.-DM
S30 SURGEON 149,– DM

S44 BARDS TALE 95.-
S62 DEJA VU 95.-DM

A48 ON LINE
150.-DM

DELUXE ART PARTS
VOL.2 - 70.-DM

DYNAMIC CAD
980,– DM

S61 WINTER GAMES 75.-
S39 WORLD GAMES 75.-DM

A54 TRUE BASIC 350.-
LEATHER GODDESSES 80,– DM

©ma

UND WEITERE
44 PROGRAMME

Porto bei Vorkasse 4.-DM
bei Nachnahme 6.-DM

PDC

Bestellservice: 9-18.3o - 06172-24748
18.3o-22 - 06171-53863

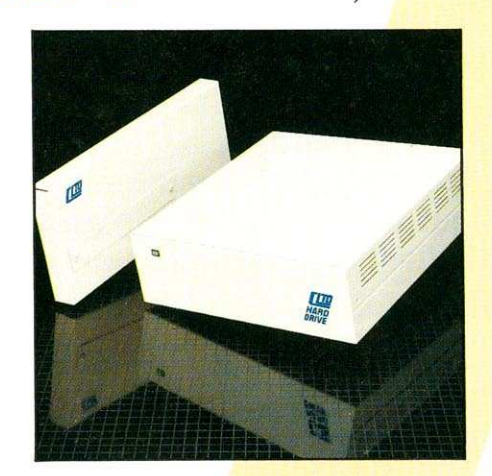
Auslandsbestellungen nur gegen
Vorkasse + 6.-DM Porto

S - SPIEL
A - ANWENDER

PDC GmbH - Louisenstr.115 - Passage Alter Bahnhof - 6380 Bad Homburg - Tel.06172-24748

20-Meg SCSI Hard Drive DM 2.495,-

- Auto-Configuration
  Durchgeschleifter Expansions-Bus
  Controller unterstützt bis zu
  sieben zusätzliche Laufwerke
- Interne Stromversorgung
- Schneller als vergleichbare Laufwerke
  durch neuen Hi-Speed Controller



50-Meg SCSI Hard-Drive DM 3495,–

Time Saver DM 179,–

- Echtzeituhr/Kalender
- mit Batteriepufferung
- 8 Kbyte programmierbares
  Constant Memory Ram
- wird in das Tastatur-
  kabel gesteckt
- voll kompatibel

JetSet

Amiga Laser Printing Software DM 149,-

- Arbeitet mit HP-kompatiblen Laser-Druckern
- Über 100 verschiedene Schriftarten erhältlich
- Arbeitet mit Textcraft & Scribble

JetSet Fonts

DM 99,- DM 199,-

Kompletter Typensatz in jedem Paket
(z.B. italic, bold italic, demi-bold, Normalschrift in verschiedenen Größen)
Die Auswahl beinhaltet:
–Times, –Triumvirate, –ITC Souvenier, –Old English, –Unical Commercial, –Script, –Dom Casual, –ITC Benguiat Bold, –Broadway, –Globe Gothic Outline, –Border Symbols, –ITC Dingbats, –ITC Souvenir Greek/Math., –ITC Times Greek/Math.
und viele, viele mehr!



- Auto Configuration.
- Arbeitet mit jeder
  Amiga Software

aMEGA Board

1 Megabyte DM 1.195,-
2 Megabyte DM 1.595,-

- Bus durchgeschleift
- Volle Garantie

Jetzt erhältlich!

Händleranfragen und Frankfurter Kunden:



☎ 069/7071102

Telef.-Kundenservice und Bestelldienst:



PDC ☎ 06172/24748
Ladenpassage Alter Bahnhof
6380 Bad Homburg

Ausdruck mit Format

AMIGA TEST Deluxe-Print bildet zusammen mit Deluxe-Video und Deluxe-Paint eine ausgezeichnete Einheit zur Grafikbearbeitung auf dem Amiga. Lesen Sie, was man mit diesem fantastischen Druckprogramm alles machen kann.

Deluxe-Print rundet die Deluxe-Serie von Electronic Arts um ein fantastisches Druckprogramm ab. Dabei können nicht nur Deluxe-Produkte Grafikdaten untereinander austauschen, sondern es können auch Grafiken anderer Programme mit Deluxe-Print aufbereitet werden. Dieses übergreifende Konzept hat sich nicht zuletzt durch die Einführung des IFF-Standards (Interchange File Format) am Markt hervorragend bewährt. Die Produkte aus der Deluxe-Serie gehören zu den meistverkauften Programmen für den Amiga. Grund genug, für uns einmal einen der bewährten Vertreter unter die Lupe zu nehmen. Grafiken, die im IFF-Standard auf einem Speichermedium vorliegen, können von den verschiedensten Programmen gelesen werden. Egal, wer das Programm herausgebracht hat, wichtig ist nur, daß IFF-Files weiterverarbeitet werden. So dürfen zum Beispiel Bilder, die mit Deluxe-Paint gemalt wurden, als Hintergrundszenen oder Objekte für Animationen mit Deluxe-Video verwendet werden. Will man vorab bestimmte Formate dieser Grafiken ausdrucken, so benutzt man die Bilder als Grundlage für Deluxe-Print. Warum kann man nicht einfach Deluxe-Paint dazu benutzen, da doch auch dieses Programm eine Druckoption hat? Diese Frage werden sich viele schon gestellt haben, doch am Ende dieses Artikels sollte jedem klar geworden sein, daß die beiden Programme sehr unterschiedlich arbeiten und sich optimal ergänzen.

Hinter Deluxe-Print steht im Prinzip die Idee, dem Benutzer ein möglichst komfortables Werkzeug an die Hand zu geben, damit er Grafiken nach Belieben ausdrucken kann. Dies bezieht sich sowohl auf verschiedene Druckgrößen, als auch auf die unterschiedlichsten Verwendungsgebiete für diese gedruckten Grafiken. So gibt es im Hauptmenü des Programms eine Auswahl zwischen sieben verschiedenen Formaten:

☐ Sign: Unter diesem Menüpunkt können Grafiken bis zur maximalen Größe einer Druckseite (8,5 x 11 Zoll) zusammengestellt werden.

☐ Label: Kann bis zu drei verschiedene kleine Bildchen (diese werden bei Deluxe-Print Images genannt) nebeneinander drucken. Dies wird vor allem für Adreß- oder Diskettenaufkleber gebraucht.

☐ Banner: Eignet sich vorwiegend für die Verwendung mit Computerendlospapier, da es Bilder und Schriftzüge im Querformat über mehrere Blätter hinweg drucken kann. Damit lassen sich wie schon der Name sagt riesige Banner oder Reklameschriftzüge erstellen.

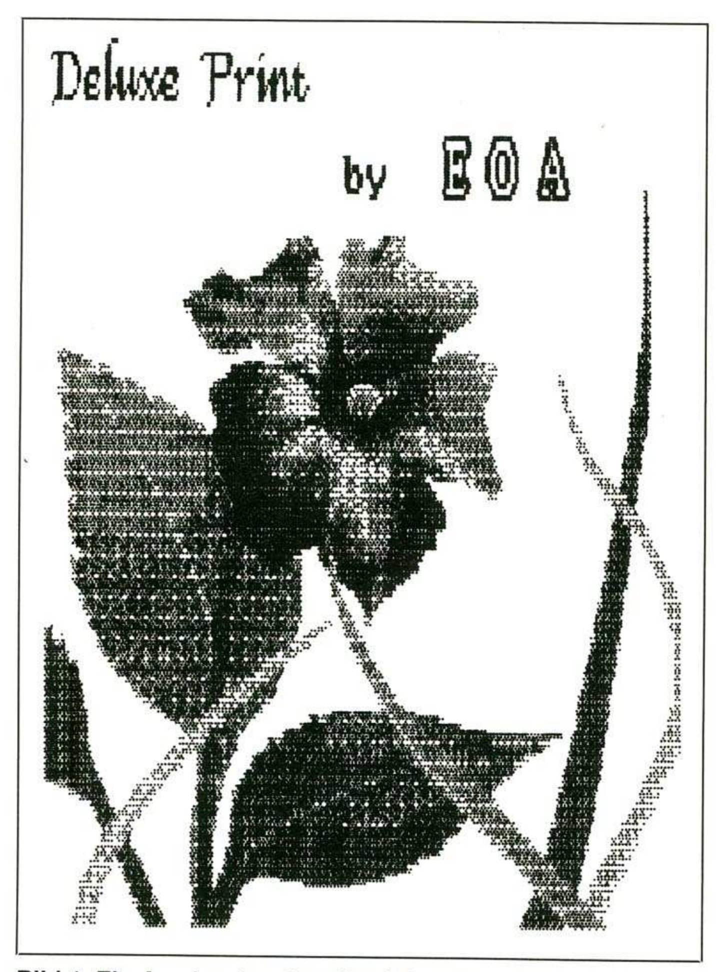


Bild 1. Ein Ausdruck unter »Semi-Custom« mit verschiedenen Schriften

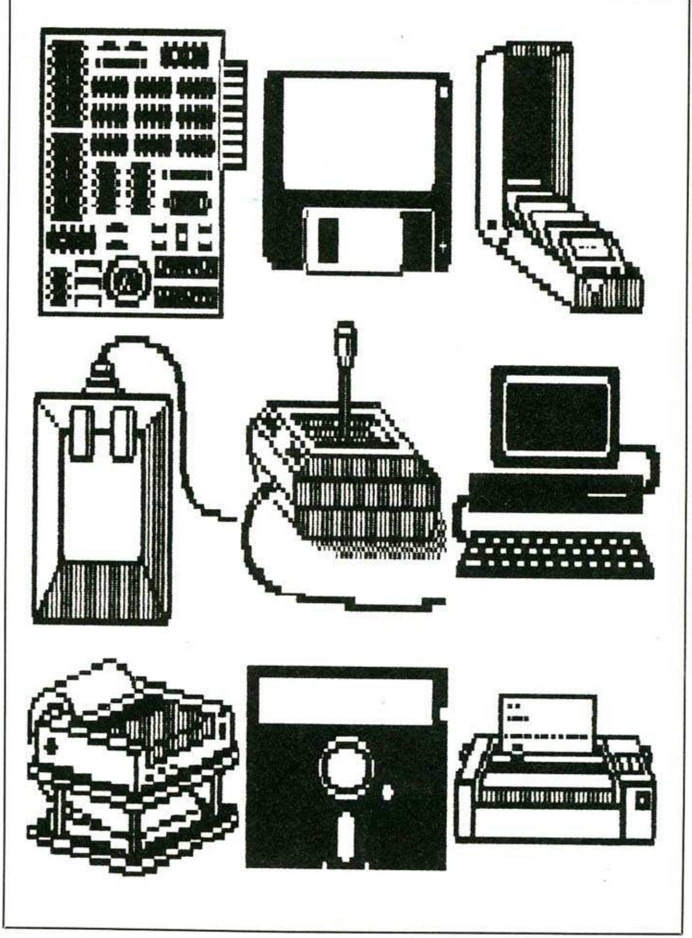


Bild 2. Es gibt eine große Zahl an guten Images für die Bildgestaltung

☐ Letterhead: Für diese Option gilt im Prinzip das gleiche wie für »Banner«. Der einzige Unterschied ist die Größe und damit auch der Verwendungszweck für die so entstandenen Grafiken. Es wird auch in einem langgezogenen Format gearbeitet, wobei der Ausdruck dann allerdings auf dem oberen Sechstel einer Seite stattfindet, um die Verwendung als Briefkopf zu ermöglichen.

☐ Calendar: Dies funktioniert ähnlich wie »Sign«, nur daß die Erstellung von Kalenderblättern wesentlich erleichtert wird.

☐ Greeting Card: Eine komplette Grußkarte kann man unter diesem Menüpunkt erstellen. Dabei wird von vier Seiten der Karte ausgegangen, die einzeln bearbeitet, später aber zusammenhängend ausgedruckt werden. Der Druck erfolgt so, daß man später das Blatt zweimal falten muß. Man erhält dann eine richtige Grußkarte mit Vorder-, Rückseite und Innenteil.

☐ 4-Tile Sign: Entspricht im wesentlichen der Grußkarte. Der Ausdruck erfolgt jedoch nicht zusammenhängend, sondern alle Bilder stehen als Viererblock nebeneinander.



Bild 3. So zeigt sich das Hauptmenü des »Sign«-Formats von Deluxe-Print

Einfache Bedienung eingeschlossen

Um die Editiermöglichkeiten, die in Deluxe-Print stecken, ausführlich zu beschreiben, stützen wir uns im folgenden auf das Format »Sign«. Darin kommen alle Funktionen voll zu Geltung. In den Untermenüs zum Editieren der anderen Formate werden diese Funktionen entsprechend verwendet.

Das Bearbeiten einer Seite im gewählten Format, hat drei Hauptmenüpunkte: Image, Border und Text. Abgesehen davon, daß man eine Grafik in bis zu 16 Farben als Hintergrund verwenden kann, bilden diese Punkte den Kern von Deluxe-Print. Die Farbanzahl ist in diesem Fall von der Auflösung abhängig in der Deluxe-Print arbeitet. In der mittleren Auflösung (Med-res) ergibt sich auf normalen 8- oder 9-Nadel-Matrixdruckern schon ein hervorragendes Abbild der entworfenen Seiten (Bild 1). Man muß natürlich in dieser Auflösung die Restriktion von 16 Farben auf einmal, sowohl auf dem Bildschirm als auch beim Anschluß von Farbdruckern, hinnehmen. Dafür kann natürlich immer noch aus der gesamten Farbpalette von 4096 Farben des Amiga geschöpft werden. Die Farben lassen sich, ähnlich wie bei Deluxe-Paint über ein Menü namens »Palette« verändern. Außerdem sind am unteren Bildschirmrand ständig alle 16 Farben zum Anklicken über die Maus vorhanden. Somit kann man schnell und effektiv einen Farbwechsel vollziehen. »Image« als einer der wichtigsten Punkte im Hauptmenü, erlaubt das Ausgestalten einer kompletten Seite mit zweifarbigen Bildern, die maximal 52 x 88 Pixel groß sein dürfen. Die Bilder sind zwar grundsätzlich nur einfarbig, es dürfen sich aber durchaus mehrere in verschiedenen Farben im Gesamtbild überlagern. Man braucht nur beim Plazieren entsprechend die aktuelle Farbe ändern. An Images wird auf der Deluxe-Print-Diskette eine ganze Anzahl an Beispielen mitgeliefert. So findet man Bildchen zu allen möglichen Themenbereichen, wie etwa Sport, Computer, Gebäude, Fahrzeuge etc. (Bild 2). Wem das noch nicht genug ist, der kann sich auf Zusatzdisketten weitere Images anschaffen. Damit man aber nicht nur auf vorgefertigte Bilder angewiesen ist, enthält das Programm einen Editor, mit dem man nach Belieben eigene Images entwerfen kann. Zusätzlich lassen sich dann schließlich noch sogenannte »Brushes« aus Deluxe-Paint als Images übernehmen (die »Brushes« sind kleine Bildschirmausschnitte, die man in Deluxe-Paint extra speichern kann). Mittels der Koordinatenanzeige von Deluxe-Paint kann man einen Brush genau so groß machen, daß er direkt als Bildchen für Deluxe-Print verwendet werden kann. Das Laden der Images, wie überhaupt der gesamte Ladevorgang für Hintergrundbilder, Rahmen und fertige Seiten, ist durch das Menü »Path« sehr komfortabel gestaltet. Darin werden die aktuellen Zugriffspfade auf die angeschlossenen Speichermedien angegeben. Auch Harddisks lassen sich auf diese Weise problemlos von Deluxe-Print aus nutzen. Die jeweiligen Files werden zuerst als Namen angezeigt und sind auf dem Bildschirm sichtbar, sobald sie selektiert wurden. Dadurch können die einzelnen Gruppen von Bildern auch schnell nach passenden Motiven durchsucht werden (Bild 3). Von den Images behält Deluxe-Print bis zu neun Stück in einer sogenannten Zwischenablage. Es sind also maximal neun verschiedene dieser Bildchen direkt und ohne Nachladen verfügbar. Ein ausgewähltes Image kann nun im Bild plaziert und auf mehrere Arten verändert werden. Innerhalb des Bildes darf es mit »Move« verschoben, mit »Flip« um jeweils 90 Grad gedreht, mit »Size« vergrößert oder verkleinert und mit »Remove« wieder entfernt werden. Hat man ein Image einmal »ergriffen«, so kann es mittels »Copy« mehrmals Verwendung finden. Natürlich bietet das Programm auch eine Vorverteilung der Images in einem Gitterraster an. Das einzige was dann noch zu tun bleibt, ist die gewünschte Verteilung auszuwählen; den Rest macht Deluxe-Print automatisch. Das gleiche gilt auch für den Menüpunkt »Text«. Mit ihr wird es überhaupt erst auf komfortable Weise möglich, Text in das Bild zu bringen. Wenn man ihn anwählt, erscheint im unteren Bildschirmteil ein Abfragefenster, in dem die gewünschten Texte einzugeben sind. Hat man den Text eingegeben, braucht nur noch mit der Maus an der gewünschten Stelle im Bild geklickt werden, und schon erscheint der Text dort. Wie schon bei den Images, muß auch hier noch nachträglich etwas verändert werden, da meist nicht alles sofort hundertprozentig stimmt. Dies ist kein Problem; auch der Text läßt sich beliebig verschieben, kopieren oder wieder entfernen. Beim Ausführen dieser Funktionen wird einem sehr schnell deutlich, wo die Unterschiede zu Deluxe-Paint liegen. Die verschiedenen Images, aber auch Texte können einzeln selektiert werden, auch wenn sie schon im Gesamtbild plaziert sind. Übereinanderliegende Texte oder Images kann das Programm mühelos auseinander halten. Auch das Erste Bildchen, welches man im Gesamtbild abgelegt hat und welches schon von anderen teilweise verdeckt wird, läßt sich so wieder entfernen. Deluxe-Paint unterstützt zwar die »Brushes«, doch ein Bild oder nur einen Hintergrund zu restaurieren, aus dem man einen eventuell falsch plazierten Brush wieder entfernt hat, dürfte oft unmöglich sein. Außerdem wird dabei ein im Bild plazierter Text in keinster Weise unterstützt. Deluxe-Print bietet zusätzlich zur Wahl verschiedener Schriftarten noch die Möglichkeit die Schrift konturiert oder im Rahmen erscheinen zu lassen. Dies ist ebenso wie das Zentrieren von Text eine Eigenschaft, die von den Produkten der Deluxe-Serie nur Deluxe-Print beherrscht.

Der Rahmen macht's

Der dritte Hauptmenüpunkt ist »Border«. Darunter versteht man das Ausstatten des Gesamtbildes mit einem Rahmen. Auch hier bietet Deluxe-Print auf Diskette einige fertige Demo-Rahmen an. Selbstverständlich kann man auch hier eigenes kreieren. Dazu dienen im wesentlichen die Images. Sie sind die Grundlage aus denen das Programm den Rahmen generiert. Die einzelnen Teile des Rahmens sind für Deluxe-Print nichts anderes als besondere Images. Teile davon werden gespiegelt und ge-

Fortsetzung auf Seite 101

Zeitung aus dem Computer

AMIGA TEST **Desktop Publishing, die komplette Zeitung aus dem Computer, ist heute in vieler Munde. In vielen Vereinen und Schulen werden eigene Zeitungen gemacht. Wird die Arbeit mit dem Programm Pagesetter auf dem Amiga leichter?**

Hinter dem Begriff Desktop Publishing verbirgt sich mehr, als man zunächst vermutet. Das Ziel aller Bemühungen ist, daß eine komplette Zeitung im Computer entsteht. Das bedeutet nicht nur, daß der Text hier eingegeben wird, sondern alle Aufgaben des Layouts bewältigt werden können. Zum Layout gehören Probleme wie die Positionierung von Bildern und Grafiken und so weiter. Wenn die Seite dann komplett fertig ist, werden die Informationen an die Satzmaschine geschickt. Inwieweit Pagesetter diese Voraussetzungen erfüllen kann, haben wir für Sie ausführlich getestet.

Nach dem Laden von Pagesetter, muß man zuerst die lästige Prozedur des Kopierschutzes über sich ergehen lassen. Es wird die genaue Position eines Wortes angegeben, das man dann aus der Anleitung heraussuchen muß. Hat man diese Hürde genommen, darf man mit der Arbeit beginnen.

Als erstes muß nun die Arbeitsgrundlage geschaffen werden, die Seite, auf der geschrieben wird. Mit Pagesetter können bis zu 99 Seiten erstellt und verwaltet werden. Einer der wichtigsten Punkte ist die Einstellung des Formats. Außer den vier vorgegebenen Formaten Standard, Legal (amerikanische Formate), DIN A4 und DIN B5 gibt es noch die Möglichkeit, die Maße direkt einzugeben. Wobei die Angabe des Formats in Inch (1 Inch = 2,54 Zentimeter) geschieht. Die kleinste einstellbare Größe ist hier zirka 5 x 2,5 cm, die maximale beträgt ungefähr 22 x 36 cm. Als nächstes können die Werte für den Abstand des Textes vom Blattrand geändert werden. Sie bestimmen, wieviel Platz auf jeder Seite des Textes frei bleiben soll. Will man den Text mehrspaltig gestalten, wie das in Zeitungen üblich ist, geht dies ohne Probleme. Man muß nur beachten, daß die Spalten nebeneinander auf die Seite passen. Hat man diese Schritte erledigt, kann man sich der weiteren Gestaltung der Seite widmen.

Wenn man das Layout der Seite bearbeitet, sieht man auf dem Bildschirm die komplette Seite oder einen Ausschnitt davon. Um genauer positionieren zu können, gibt es einige Hilfsmittel. So können die Koordinaten des Mauszeigers angezeigt werden. Bei den beiden anderen Möglichkeiten, ein Lineal und ein Gitter, ist die Einteilung wählbar. Die Einheiten hierfür sind Inch oder Pica (eine Schriftart). Mit einem weiteren Befehl kann man erreichen, daß alle Elemente auf der Seite sich automatisch an das Raster anpassen. Dadurch erspart man sich die Kleinarbeit, alles per Hand genau zu positionieren.

Pagesetter hat drei verschiedene Modi, in denen die Seite dargestellt wird. Im ersten Modus ist der Text nicht lesbar, da die gesamte Seite angezeigt wird, aber eine grobe Aufteilung der Seite ist hier am besten möglich. Mit der Zoom-Funktion kann man die Größe des Ausschnitts bestimmen, der sichtbar ist. Befindet man sich in der nächsten Stufe, werden zirka $\frac{2}{3}$ der Gesamtseite angezeigt. Hier lassen sich die verschiedenen Größen schon recht genau einstellen. Im letzten Modus wird ungefähr $\frac{1}{8}$ der Seite gezeigt (siehe Bild 1). Der Text ist jetzt lesbar, und alle Details, die man vorher vielleicht übersehen hat, treten jetzt hier ganz klar zu Tage.

Bild 1. Ein Ausschnitt der Seite mit Grafik und Text

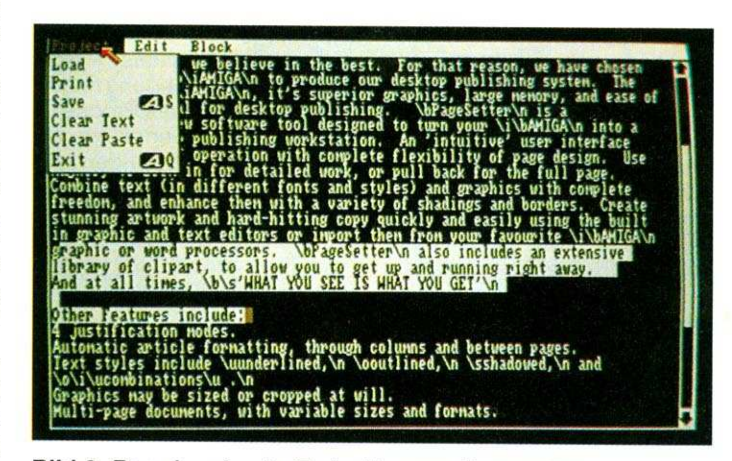


Bild 2. Der eingebaute Texteditor von Pagesetter

Bei Pagesetter wird nicht direkt auf die erstellte Seite geschrieben, sondern in sogenannte Boxen. Diese Boxen sind sozusagen Schachteln, die man mit Text oder Grafik füllen kann. Für eine Seite werden normalerweise einige Boxen benötigt, um Grafiken und Spalten zu erzeugen. Die Schachteln sind dann beliebig auf der Seite zu verschieben und können auch auf andere Seiten übertragen werden. Die beste Vorgehensweise ist, die Box zunächst sehr groß zu machen und sie dann auf die durch den Inhalt benötigte Größe zu bringen.

Eine der hervorragendsten Eigenschaften des Programms ist die für Textboxen mögliche Verkettung. Wenn Sie einen Text schreiben, der für die erste Box zu lang ist, wird er automatisch in die zweite Box eingetragen. Auf diese Weise ist es möglich, auch sehr lange Texte über mehrere Boxen und auch Seiten zu schreiben. Entfernt man nun eine Box aus dieser Kette, geht der Inhalt dennoch nicht verloren, er wird einfach in der darauffolgenden fortgesetzt. Die einzige Box, die man nicht ohne weiteres entfernen darf, ist die erste in der Kette, da der Text dann auch verloren ist.

Für jede Box, die man erzeugt, lassen sich die Parameter getrennt einstellen. So ist wählbar, ob eine Schachtel transparent oder undurchsichtig ist. Wichtig wird dies, wenn zwei Boxen übereinanderliegen. Ist die vordere transparent, sieht man den Inhalt der hinteren auch, ansonsten wird er verdeckt.

Des weiteren lassen sich vier verschiedene Arten der Formatierung anwählen, Blocksatz, links- und rechtsbündiger Flattersatz sowie zentrierte Darstellung. Normalerweise wird der Text bis an die Ränder geschrieben. Durch Wählen der Werte für oberen und linken Randabstand, läßt sich dies jedoch genau wie bei der Seite ändern.

Das wichtigste ist aber die Einstellung des gewünschten Zeichensatzes, wobei alle für den Amiga verfügbaren Fonts verwendet werden können. Für die Textdarstellung ist wichtig, mit welchem Zeilen- und Zeichenabstand gearbeitet wird. Es gibt auch noch zwei Einstel-

lungen, die der grafischen Gestaltung dienen. Die erste ist die Wahl des Hintergrundes, wobei verschiedene Muster hinterlegt werden können. Bei der Gestaltung eines Rahmens, stehen viele Möglichkeiten offen, außer der Dicke der Umrandung kann noch eingestellt werden, ob sie rundherum geht oder nur auf zwei Seiten gezeichnet wird. Einige Beispiele sehen Sie in Bild 4.

Nachdem wir besprochen haben, wie Text auf der Seite verwaltet wird, kommen wir zur Erstellung des Textes. Hierzu dient ein eigener Texteditor (siehe Bild 2), der vollkommen unabhängig von der grafischen Darstellung der Seite arbeitet. Man arbeitet wie bei anderen Textverarbeitungen auf dem Amiga mit Pull-Down-Menüs und der Maus. Die Erstellung eines Textes ist komfortabel möglich und auch das nachträgliche Editieren wird gut unterstützt. Ein Punkt ist besonders interessant: Man kann Texte von anderen Textverarbeitungen wie Textcraft, Scribble und Vizawrite laden und weiterbearbeiten. Das ermöglicht das Erstellen des Textes mit einem Editor, der darauf spezialisiert ist und noch mehr Möglichkeiten bietet.

Ist das Schriftstück dann fertig, verläßt man den Editor einfach, ohne sich um die Formatierung zu kümmern. Wenn man nun die Seite mit den Boxen vor sich sieht, füllt man die gewünschte Box mit dem Text.

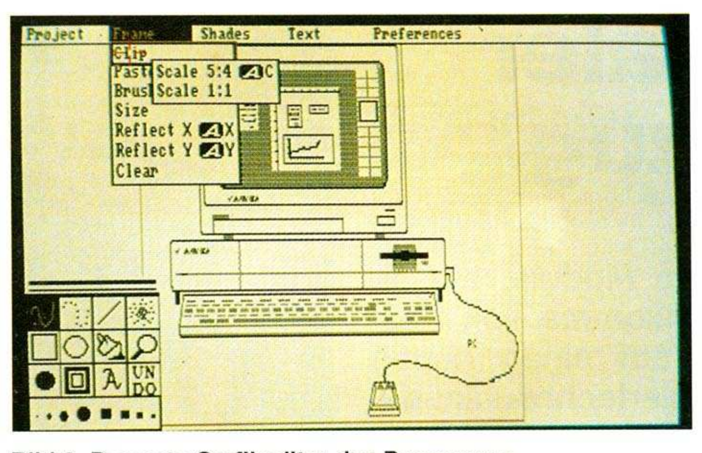


Bild 3. Der gute Grafikeditor des Programms

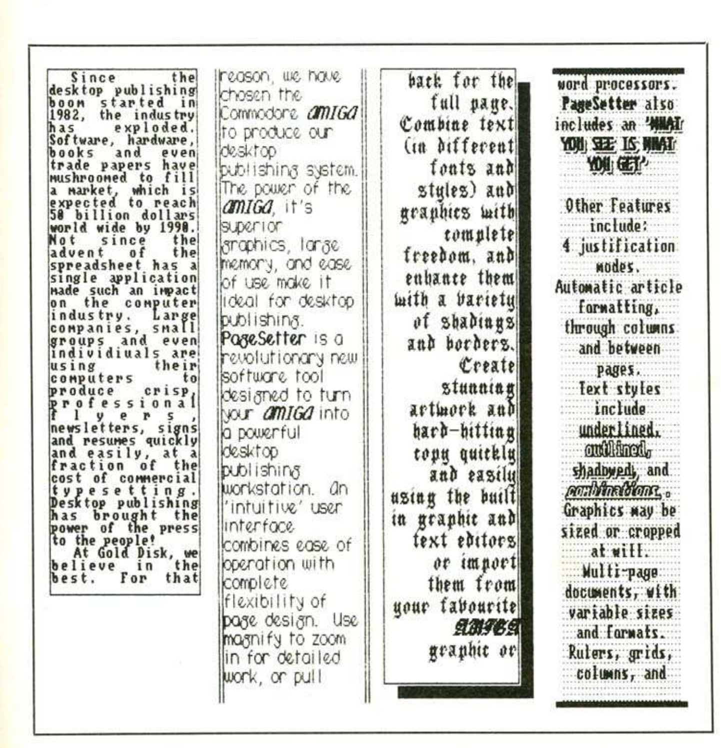
Das Aussehen des Textes kann dann auch noch mit der Einstellung der Box-Parameter erfolgen. Entdeckt man noch Fehler im Text, ist der Text beliebig oft mit dem Editor veränderbar. Da das Umschalten von der Ansicht der Seite in den Texteditor jedoch etwas Zeit kostet, gibt es noch eine andere Möglichkeit kurze Texte in eine der Schachteln einzutragen. Dabei wird nicht in die Textverarbeitung umgeschaltet, sondern es erscheint nur ein kleines Fenster, was natürlich viel Zeit spart.

Doch was wäre eine Zeitung ohne Bilder? Auch in diesem Bereich kann Pagesetter überzeugen. Wie beim Texteditor, wird die Grafik nicht direkt auf der Seite gemalt, sondern in einem speziellen Grafikeditor mit eigenem Bildschirm (siehe Bild 3). Mit den vorhandenen Befehlen lassen sich auch ohne weiteres komplexe Grafiken zeichnen. Eines der mitgelieferten Bilder ist in Bild 5 zu sehen. Bilder, die im IFF-Format vorliegen und eine Auflösung von 320 x 200 oder 640 x 200 Punkten besitzen, lassen sich auch laden. Bei mehrfarbigen Zeichnungen, die zum Beispiel mit DPaint II gemalt wurden, werden die Farben in verschiedene Punktmuster umgesetzt. Wie gut das Ergebnis dieser Konvertierung wird, ist unter anderem davon abhängig, wie groß die verschiedenfarbigen Flächen sind.

Zum Übertragen des Bildes auf die Seite, verfährt man ähnlich wie im Texteditor. Zuerst muß der gewünschte rechteckige Bereich aus dem Bildschirm ausgeschnitten werden, dann verläßt man den Grafikeditor und legt die Grafik in einer Box ab. Ist man mit dem Ergebnis nicht zufrieden, läßt sich das Bild im Grafikeditor wieder verändern und erneut in die Box einfügen. Das punktgenaue Justieren der Grafik auf der Seite ist dann sehr leicht zu bewerkstelligen, ohne die Box dabei zu verschieben.

Alles in allem lassen sich mit Pagesetter recht schnell und einfach Zeitungsseiten erstellen. Da die Qualität des Ausdrucks jedoch nicht ganz überzeugen kann, was aber am Druckertreiber liegt, dürfte der Anwendungsbereich eher auf Schüler- und Vereinszeitungen sowie den Heimbereich beschränkt sein. Von professionellem Desktop Publishing kann hier also nicht die Rede sein, obwohl schon ein Schritt in die richtige Richtung getan ist. Pagesetter ist das erste Programm auf dem Amiga, das in dieser Richtung existiert und man kann hoffen, daß in Zukunft noch einiges in dieser Beziehung erscheinen wird. Für denjenigen, der kein absolut professionelles Ergebnis erwartet, ist die Anschaffung aber sicher sinnvoll. (rb)

Bezugsquellen:
Interplan GmbH, 8000 München 19, Nymphenburger Straße 134, Tel. (089) 1234066
Philgerma GmbH & Co.KG, 8000 München 40, Ungererstraße 42, Preis: 378 Mark
Softwareland, Schweiz, 8050 Zürich, Franklinstraße 27, Preis: 349 Mark
Soltau Electronics GmbH & Co.KG, 2000 Hamburg 36, Esplanade 39, Preis: 349 Mark



Since the desktop publishing boom started in 1982, the industry has exploded. Software, hardware, books and even trade papers have mushroomed to fill a market, which is expected to reach 58 billion dollars world wide by 1990. Not since the advent of the spreadsheet has a single application made such an impact on the computer industry. Large companies, small groups and even individiuals are using their computers to produce crisp, professional flyers, newsletters, signs and resumes quickly and easily, at a fraction of the cost of commercial typesetting. Desktop publishing has brought the power of the press to the people!
At Gold Disk, we believe in the best. For that

reason, we have chosen the Commodore AMIGA to produce our desktop publishing system. The power of the AMIGA, it's superior graphics, large memory, and ease of use make it ideal for desktop publishing. PageSetter is a revolutionary new software tool designed to turn your AMIGA into a powerful desktop publishing workstation. An 'intuitive' user interface combines ease of operation with complete flexibility of page design. Use magnify to zoom in for detailed work, or pull

back for the full page. Combine text (in different fonts and styles) and graphics with complete freedom, and enhance them with a variety of shadings and borders. Create stunning artwork and hard-hitting copy quickly and easily using the built in graphic and text editors or import them from your favourite AMIGA graphic or

word processors. PageSetter also includes an 'WHAT YOU SEE IS WHAT YOU GET'.
Other Features include:
4 justification modes.
Automatic article formatting, through columns and between pages.
Text styles include underlined, outlined, shadowed, and combinations.
Graphics may be sized or cropped at will.
Multi-page documents, with variable sizes and formats.
Rulers, grids, columns, and

Bild 4. Vier Boxen mit verschiedenen Parametern für Hintergrund, Zeichensatz und Rahmen

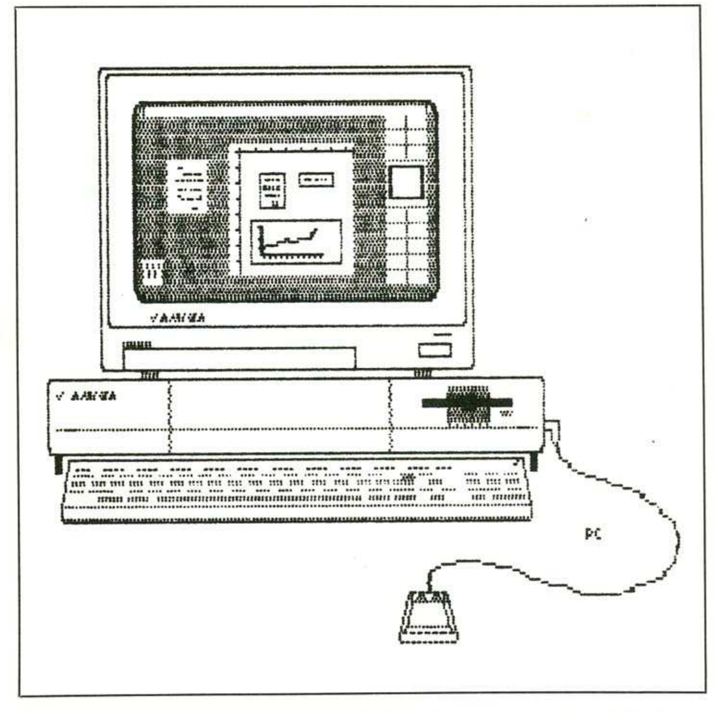


Bild 5. Eine der mitgelieferten Grafiken als Ausdruck mit einem Epson FX-80

Ein genialer Schachzug



Der Modula-Compiler von TDI unterstützt den Amiga in vollem Umfang

AMIGA TEST **Die Sprache Modula, ein direkter Nachkomme von Pascal, weist auf einen neuen Trend in der Programmiertechnik hin: die Modularisierung. Mit dem Modula-2-Compiler von TDI Software steht nun auch für den Amiga diese Sprache der Zukunft zur Verfügung. Wir haben Modula für Sie etwas durchleuchtet und den Compiler ausführlich getestet.**

Mit der Sprache Pascal, die auch heute noch im PC-Bereich dominierend ist, wurde der Grundstock für einen neuen Programmierstil gelegt. Pascal ermöglichte als erste Sprache konkret die Erstellung klarer und strukturierter Programme. Doch der Entwickler Professor Wirth ruhte nicht und schuf Anfang der achziger Jahre ein neues Werk: Modula, das den Begriff der strukturierten Programmierung sehr viel stärker verwirklicht, als vergleichbare Sprachen wie Pascal oder gar C.

Der Name dieser Sprache weist bereits auf ihr grundlegendes Konzept hin. Modula unterstützt das Prinzip der Modularisierung von Programmen. Was das bedeutet, ist schnell erklärt:

Soll ein Problem programmiert werden, ist es vorteilhaft, die Gesamtaufgabe in viele kleine, möglichst allgemeine Teile zu zerlegen. Kleine Probleme lassen sich sehr viel leichter lösen als große. Hat man die Teilprogramme fertiggestellt, genügt es schließlich, diese in einer logischen Form zusammenzusetzen, um das gesamte zu programmierende Problem zu lösen. Diese Methode hat den Vorteil, daß die Teilprogramme (Fachwort: Subroutinen) in einem anderen Programm wieder verwendet werden können, sollte ein ähnliches Problem anfallen. Modula behandelt solche Unterprogramme als Module, die in sich abgeschlossen, vollkommen unabhängig arbeiten können. Ein Modula-Programm setzt sich aus solchen Modulen zusammen. Soll ein Datenaustausch zwischen zwei Teilen stattfinden, kann dies nur über streng festgelegte Schnittstellen (formale Parameter) erfolgen. Ein aus Modulen bestehendes Programm wird schließlich selbst zum Modul, das wiederum als Teilmodul in anderen Programmen wirken kann. Das heißt aber nun nicht, daß immer das komplette Modul in das neue Programm eingebunden wird. Beim Aufruf des Moduls können Sie viel mehr angeben, welche Prozeduren oder Funktionen aus dem Modul übernommen werden.

Ein neuer Begriff: Modularisierung

Der Sprachkern von Modula umfaßt nur relativ wenig Befehlselemente. Sie sind denen von Pascal sehr ähnlich. So müssen auch hier Konstanten und Variablen vor Gebrauch deklariert werden. Strukturanweisungen wie WHILE DO, REPEAT..UNTIL und CASE zwingen innerhalb eines Moduls zu einem klaren Aufbau.

Die meisten Befehle stehen in Form von Prozeduren in Modulen bereit, aus denen die benötigten Teile in das eigene Programm »importiert« werden. Dort findet man beispielsweise Routinen zur Ein- und Ausgabe auf dem Bildschirm oder Rechenfunktionen wie SIN oder LOG.

Modula kennt prinzipiell zwei Arten von Modulen: die Empfänger-Module (»Client-Moduls«) und die Bibliotheks-Module (»Library-Moduls«). Empfänger-Module »importieren« in der Regel nur Teile aus anderen Modulen und sind nicht dazu gedacht, Werte oder Routinen zu »exportieren«, das heißt an andere Module weiterzugeben. Ein Modula-Hauptprogramm wird deshalb immer ein Empfänger-Modul sein. Es verwendet nur bestehende Module und enthält normalerweise keine eigenen Unterroutinen. Bibliotheks-Module »exportieren« dagegen hauptsächlich Teile ihrer selbst, können allerdings Teile aus anderen Modulen für die eigene Arbeit übernehmen. Sie sind quasi die Unterprogramme eines Hauptprogramms. Ein solches Modul besteht aus zwei Teilen.

Das »definition module« stellt die Schnittstelle zwischen der Bibliothek und dem Haupt-Modul dar. Es enthält sämtliche Konstanten, Variablen und »Importe« aus anderen Modulen, die verwendet werden. Der ausführende Teil des Modules ist das »implementation module«. Es enthält die eigentlichen Anweisungen zur Bewältigung des entsprechenden Teilproblems.

Will man nun ein Programm in Modula schreiben, wird man das zu programmierende Problem zunächst in kleine Teile zerlegen. Die Einzelprobleme werden anschließend als Bibliotheks-Module definiert, sofern sie sich nicht bereits durch frühere Programmierungen in der Modulsammlung befinden. Schließlich entwirft man das Empfänger-Modul, das die benötigten Module importiert und in einer entsprechenden Reihenfolge verbindet.

Modula mit dem Amiga

Mit TDI Modula-2, das in der Zwischenzeit in Version 3.00a ausgeliefert wird, kann nun auch der Amiga-Besitzer in Modula programmieren. Bevor man jedoch eigene Programme auf dem Amiga schreiben kann, ist es wichtig, sich mit dem Modula-2-System vertraut zu machen. Geliefert wird der Compiler auf zwei Disketten zusammen mit einem etwa 300 Seiten umfassenden Handbuch. Da es sich, wie die meiste Software für den Amiga, um ein englisches Produkt handelt, sind bei der Lektüre der Anleitung Englischkenntnisse erforderlich. Mit etwas Schulenglisch und einem englischen Wörterbuch kommt man jedoch gut zurecht, zumal der anleitende Teil des Handbuches recht kurz ist. Der größte Teil der 300 Seiten befaßt sich mit den mitgelieferten Modulen, während lediglich knapp 50 Seiten das Modula-2-System erklären. Als Modula-Lehrbuch kann es also nicht verstanden werden. Teilweise erreicht man das gewünschte Ergebnis nur durch Austesten des Compilers. Das Handbuch könnte also durchaus komfortabler und ausführlicher gestaltet sein.

Bereits auf den ersten Seiten wird der Leser aufgefordert, Sicherheitskopien der beiden Disketten anzufertigen, was problemlos zu bewerkstelligen ist, da TDI Modula-2 keinen Kopierschutz besitzt. Hat man die Originale an einem sicheren Ort verwahrt, kann mit den Vorbereitungen begonnen werden. Ist man Besitzer von zwei Diskettenlaufwerken, können die Disketten in der mitgelieferten Form verwendet werden. Bei nur einer Diskettenstation müssen alle benötigten Teile auf eine Diskette umkopiert werden. Das sind:

— der Editor
— der Compiler
— der Linker
— die entsprechenden Teile der Modul-Bibliothek
— benötigte Teile der CLI von der Workbench.

Sollte die Modul-Sammlung wachsen, können dabei Speicherplatzprobleme auftreten. Daher ist die Verwendung von zwei Floppystationen ratsam, wenn nicht sogar zwingend notwendig.

Für die Arbeit mit Modula-2 sollten Sie mit der Arbeitsweise des CLI (Command Line Interface) des Amiga vertraut sein, da sämtliche Teile von dort aus aufgerufen werden. Der erste Schritt zum perfekten Modula-Programm ist die Eingabe des Programmtextes, welche mit einem komfortablen Fullscreen-Editor erfolgt. Wird von der Amiga-Systemebene der Editor aufgerufen, erscheint nach kurzer Zeit ein Window mit einer Kommandoleiste in der obersten Bildschirmzeile. Der gesamte Bildschirm dient dabei zur Programmeingabe. Mit der Maus können nun von der Kommandozeile die verschiedensten Befehle zur Editierung des Textes erreicht werden. Der Programmierer hat hier von einfachem Löschen eines bestimmten Blocks bis hin zum Verschieben ganzer Programmteile gute Werkzeuge in der Hand, um sein Programm schnell und einfach zu bearbeiten. Im Editor selbst kann das Programm bereits nach Fehlern durchsucht werden, bis es schließlich in perfekter Form dem Compiler übergeben werden kann. Nach Speichern des Textes kann der Editor verlassen werden. Nun wird der Compiler aufgerufen:

```
modula (Programmname)
gegebenenfalls Optionen
```

Der Compiler kennt zwei Anweisungen, die das Erstellen eines Listing-Files und die Ausgabe der importierten Module während der Compilation ermöglichen. Weitere Compiler-Direktiven können im Modula-Programm selbst angegeben werden. Mit deren Hilfe lassen sich verschiedenste Kontrollen für den späteren Programmablauf einstellen. Beispiele sind Prüfroutinen für den Programmstack oder Grenztests bei Feldvariablen. Trotz der Fehlersuche im Editor können während der Compilation Fehler auftreten, wenn sich etwa ein benötigtes File nicht auf der Diskette befindet. War der Übersetzungsvorgang erfolgreich, ist das entstandene Compilat allerdings noch nicht lauffähig.

Es muß dem Linker übergeben werden, der die nötigen Prozeduren aus den Modulen an das Programm bindet. Eine einstellbare Optimierung veranlaßt den Linker, eventuell definierte Routinen, die jedoch keinen Bezug zu jeglichen Programmteilen haben, zu entfernen, so daß ein kompakteres Compilat entsteht. Das fertige Programm wird schließlich unter seinem Modulnamen auf der Diskette abgelegt, und kann von der Amiga-Systemebene geladen werden.

Das Modula-2-System von TDI enthält sämtliche Bibliotheks-Module, die Niklaus Wirth in seinem Buch »Programmieren in Modula-2« als Standard voraussetzt.

Erweiterungen über Erweiterungen

TDI Modula-2 hält aber Erweiterungen bereit. Speziell für den Amiga verfaßte Module erlauben die direkte Arbeit mit dem ROM-Kernel, der Intuition, der Workbench und dem Amiga-DOS. Die Grafik des Amiga wird ebenfalls durch ein Modul unterstützt. Vordefinierte Konstante, wie NULL oder BITSPERBYTE, sowie neue Datentypen sind verfügbar. Daneben erlauben die Prozeduren MIN und MAX die Ermittlung der kleinsten oder größten Zahl einer Gruppe von Variablen.

Eine Besonderheit ist die Verarbeitung von 64-Bit-Fließkommazahlen, die sich im Bereich von 3.0E-308 bis 3.0E+308 bewegen dürfen.

Obwohl der Compiler und der Linker von TDI Modula-2 einen kompakten und recht schnellen Code erzeugen, kann der Maschinensprache-Freak mit Hilfe der Anweisung CODE 68000-Maschinencode direkt in ein Modula-Programm einbinden.

Standard oder professionell

TDI bietet sein Modula-2-Paket in drei Versionen an. Die Standard-Version beinhaltet das gesamte Compiler-System und ein englisches Handbuch. Das Entwickler-Paket gibt dem Programmierer umfangreiche Programmierhilfen. Dabei handelt es sich um einen Decoder für Link-, Symbol-, und Load-Dateien und einen Cross-Reference-Lister. Zusätzlich findet man auf einer der beiden Disketten den Quelltext zum »Runtime-Support-Modul« (AMIGAX.DEF), das im Handbuch nicht dokumentiert ist. Die kommerzielle Version schließlich beinhaltet zusätzlich die Quelltexte der Implementations-Module der Bibliothek. Um sinnvolle Anwendungen schreiben zu können, benötigt man in jedem Fall die mittlere Version.

Wollen Sie mit dem Amiga in die neue Zukunft der Programmiersprachen einsteigen, wird Ihnen Modula-2 von TDI, ob in der einen oder anderen Version, bestens behilflich sein. Da Modula der Sprache Pascal sehr ähnlich ist, werden sich Pascal-Kenner bei der Einarbeitung in diese moderne Programmiersprache sehr leicht tun. Eine gute Geldanlage, denn Modula wird in Zukunft sicherlich von sich reden machen. Die Frage ist allerdings, ob Modula gegen die auf dem Amiga neben Basic vorherrschende Sprache C ankommen kann, da mit Hilfe von C leichter auf die Betriebssystemroutinen zugegriffen werden kann. Schließlich wurde das Betriebssystem in C programmiert. Für kommerzielle Anwendungen aber, bei denen man nicht in vollem Umfang auf das Betriebssystem angewiesen ist, stellt Modula mit Sicherheit eine sinnvolle und komfortable Alternative dar. Durch das von Modula verwendete Modul-Konzept kann sich der Programmierer, ähnlich wie bei den Makros in Assembler, eine eigene Programmiersprache schaffen, die den Programmierer in komfortabler Form unterstützt.

(Michael Thomas/rf)

Philgerma GmbH, Ungererstr. 42, 8000 München 40, TDI-Modula 2 Standard, 298 Mark, TDI Modula 2 für Entwickler, ca. 350 Mark
Softwareland, Franklinstr. 27, CH-8050 Zürich, TDI Modula 2 Standardversion, 209 Mark, TDI Modula 2 Entwickler, 349, TDI Modula 2 Kommerzielle Version, 689 Mark

Fortsetzung von Seite 97

dreht damit ein gleichmäßiger Rahmen um das Bild zustande kommt.

Die zentrale Funktion und Aufgabe von Deluxe-Print ist natürlich immer noch das Ausdrucken der erstellten Bilder. Man möchte die schönen Grafiken ja nicht nur am Bildschirm betrachten, sondern vor allem zu Papier bringen, um sie als Grußkarten zu versenden, oder einfach um sie Anderen zu zeigen. Bei Aufruf von »Print« aus dem Hauptmenü erscheint das »Printer Control Panel« von dem aus man den Ausdruck auf drei verschiedene Arten starten kann. Die erste und einfachste Art ist dabei das Feld »Automatic« anzuklicken. Danach braucht man sich um nichts weiter zu kümmern. Vorausgesetzt man hat den Drucker richtig angeschlossen, eingestellt und über die »Preferences« am Amiga den richtigen Drucker gewählt. Sollte dies der Fall sein so druckt das Programm automatisch mit seinen Standardwerten. Dies wäre zum Beispiel bei »Sign« eine Größe von 8,5 x 11 Zoll. Wer die Größe des Ausdrucks nach eigenem Geschmack beeinflussen will, für den gibt es die restlichen zwei Druckformate »Semi-Auto« und »Custom«. Die einfachere von beiden ist »Semi-Auto«, bei der man zusätzlich als Zahlenwerte die gewünschte Breite des Ausdrucks in Spalten und die Höhe in Zeilen angeben kann. Deluxe-Print paßt bei der Eingabe der Zahlenwerte den Ausdruck automatisch proportional an. Das Bild wird zwar entsprechend kleiner, Verzerrungen treten jedoch nicht auf. Bei »Custom« muß man die Größe des Ausdrucks sogar in Einzelpunkten bestimmen. Je nach Angabe kann es hierbei zu Verzerrungen in der Höhe oder Breite kommen. Im Vergleich zu anderen Programmen muß man sagen, daß Deluxe-Print bezüglich des Drucks sehr komfortabel ausgestattet ist. Einziger Kritikpunkt bei der ganzen Angelegenheit ist eigentlich nur, daß man für einige Zeichensätze auf die deutschen Umlaute verzichten muß. Dies kann man aber nicht direkt Deluxe-Print anrechnen. Neue Zeichensätze werden dieses Problem in Zukunft hoffentlich beseitigen. Auf dem Amiga bildet Deluxe-Print zusammen mit den anderen Deluxe-Programmen zur Zeit allerdings eine unschlagbare Einheit. (jk)

Zing! Ein Werkzeug mit Pfiff

AMIGA TEST

Am CLI des Amiga kommt kein ernsthafter Anwender vorbei. Doch das Arbeiten auf dieser systemnahen Befehlsebene erfordert genaue Kenntnis der Befehlsstruktur. Zing! verwaltet spielend leicht das gesamte CLI und beschleunigt zudem die Diskettenarbeit. Außerdem vereinfacht es Operationen mit Diskettendateien.

Intuition ist als Benutzeroberfläche des Amiga ein für Einsteiger sehr hilfreiches und für den normalen Umgang mit Programmen absolut narrensicheres System. Es hilft durch mausunterstützte Menü- und Iconverwaltung den Amiga spielend leicht zu bedienen. Wer einmal mit einer so komfortablen Benutzeroberfläche gearbeitet hat, möchte sie nie mehr missen. Für einige Arbeiten jedoch ist eine solche Benutzeroberfläche nur ungenügend zu gebrauchen. Möchte man zum Beispiel auf der Diskette Manipulationen an bestimmten Files vornehmen; sei es, um sie zu löschen oder um ihren Inhalt auf dem Bildschirm aufzulisten, so wird man schnell an die Grenzen von Intuition stoßen. »Warum?« werden Sie sich fragen, denn man kann doch Files auch von Intuition aus löschen (DISCARD) und mit dem Notepad lassen sich auch die Inhalte von Files anzeigen. Das DOS (Disk Operating System) des Amiga und auch das gesamte Betriebssystem kennt jedoch unter anderem auch Files, die kein Icon haben und somit auf der Workbench gar nicht sichtbar sind. Ein solches File könnte man nur vom CLI aus sichtbar machen (DIR) und löschen (DELETE). Das CLI stellt aber nicht nur das DOS des Amiga dar, sondern es ist mehr als nur zur Diskettenarbeit zu gebrauchen und kein ernsthafter Amiga-Besitzer wird über kurz oder lang daran vorbeikommen. Schwierig ist nur die Syntax einiger Befehle, das umständliche Eintippen dieser Befehle am Bildschirm aber auch der nicht gerade übersichtliche Aufbau der Diskettenverzeichnisse des Amiga-DOS. Jeder der sich etwas damit beschäftigt hat, wird den Dschungel der Directory-Pfade, das Umstellen auf andere Laufwerke und die Fehlermeldung »Please replace volume ... in any drive« schon verflucht haben. Zing! ist das CLI-Utility, das mit diesen Verhältnissen gründlich aufräumt.

Das Zing!-System besteht aus mehreren einzelnen Programmen oder Modulen, die bei Bedarf aufgerufen werden.

Ein variables System

Einige davon sind auch nichts weiter als CLI-Befehle und sie befinden sich in der Subdirectory »c« auf der Diskette, sowie alle anderen CLI-Befehle. Das Besondere daran ist jedoch, daß sie wesentlich schneller arbeiten als die normalen CLI-Befehle, die ähnliches bewirken. Für alle die meinen, daß es nun schon wieder los geht mit dem Lernen der neuen Befehle, wo man doch die alten gerade verstanden hat, denen sei gesagt, daß Zing! ähnlich einer Benutzeroberfläche alle zusätzlichen Befehle leicht verwaltbar und übersichtlich macht. Außerdem wird der Anwender außer dem Geschwindigkeitsgewinn nichts direkt davon bemerken. Bevor wir zum Hauptteil des gesamten Systems kommen, dem »File System Window«, wollen wir noch die restlichen Software-Werkzeuge ansprechen, die Zing! fast nebenbei noch zur Verfügung stellt.

Viele Tools in einem

Da Zing! als Ganzes gesehen ein modulares System ist, bei dem die benötigten Programmteile nur bei Bedarf in den Speicher geladen werden und nach Gebrauch wieder verschwinden, ist eine enge Verwandtschaft zum CLI selbst nicht zu verschweigen. Das spart Speicherplatz und ermöglicht zudem das Zing!-System im Hintergrund zu installieren, während man im Vordergrund beispielsweise in einer Textverarbeitung gerade einen Brief schreibt. Das einzige, was dabei Speicherplatz verbraucht, ist ein Programm, das dem Benutzer die Möglichkeit gibt, durch die Kombination von linker Amiga-Taste und einer Funktionstaste die jeweilige Zing!-Anwendung auf den Bildschirm zu holen. Ein weiterer Vorteil davon ist, daß man Zing! auch auf anderen Speichermedien, sprich Festplatten, mehrere Floppylaufwerke oder im RAM, installieren kann. Dies geschieht einfach mit dem »ASSIGN«-Befehl.

Nachdem Zing! geladen wurde, sieht man zunächst nichts weiter als ein kleines Amiga-DOS-Fenster im unteren linken Bildschirmteil. Um überhaupt einen Einstieg zu finden, muß man nun die linke Amiga-Taste und <F10> drücken. Dies bringt das Hilfsfenster des Zing!-Gesamtsystems auf den Bildschirm (Bild 1). Darin werden die einzelnen Programmteile und die entsprechende Tastenkombination aus linker Amiga- und F-Taste aufgeführt, um die einzelnen Module jederzeit aus den verschiedensten Programmen heraus anzuspringen. Dies sind im einzelnen:

- File System Window: Dieser Hauptteil von Zing! wird ausführlich nach der Übersicht besprochen.
- New CLI: Öffnet ein neues CLI-Fenster

Bild 1. Der Bildschirm nach dem Installieren und Aufrufen des Help-Screen



Bild 2. So zeigt sich das File System Window mit maximaler Übersicht

»Happy-Computer« bringt Ihr Computer-Hobby in Schwung!



DAS GROSSE HEIMCOMPUTER-MAGAZIN MIT DEM RIESEN SPIELE-SONDERTEIL

HAPPY-COMPUTER

»Happy-Computer« berichtet über News und Facts, Trends und Preise, technische Details, Qualität und Quantität des gesamten Home-Computer-Angebots.
»Happy-Computer« vergleicht, testet, informiert.

Mit »Happy-Computer« wird der Markt für Sie transparent.

Zum vorteilhaften Abonnementpreis erhalten Sie »Happy-Computer« Monat für Monat druckfrisch und pünktlich ins Haus. Füllen Sie dazu das nebenstehende Kennenlern-Angebot aus, und senden Sie es an Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar bei München.

KENNENLERN-ANGEBOT MIT KOSTENLOSEM PROBEEXEMPLAR VON »HAPPY-COMPUTER«

☐ Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »Happy-Computer« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »Happy-Computer« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur 66,– DM statt 72,– DM Einzelverkaufspreis (Ausland auf Anfrage).

Vorname, Name

Straße, Nr.

PLZ, Ort

Datum, 1. Unterschrift

Mir ist bekannt, daß ich meine Bestellung innerhalb von 8 Tagen nach Erhalt des Probeexemplares widerrufen kann. Die rechtzeitige Absendung an Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Str. 2, 8013 Haar genügt. Ich bestätige dies durch meine 2. Unterschrift.

Datum, 2. Unterschrift

A6/7/87

☐ Save screen to IFF: Bringt ein Abfragefenster zum Vorschein zur Eingabe eines Filenamens, unter dem der Bildschirm auf Diskette gespeichert werden soll. Es wird natürlich das Bild zum Zeitpunkt des Tastendrucks genommen und das Format auf Diskette ist IFF; kann also von Zeichenprogrammen weiterverarbeitet werden.

☐ Save screen to Printer: Funktioniert im Prinzip ähnlich wie Punkt 3, nur daß dabei der Bildschirm wahlweise in drei Größen auf dem Drucker ausgegeben werden kann.

☐ Increment dimmer interval: Solange Zing! installiert ist, wird nach einer voreingestellten Zeit von fünf Minuten der Bildschirm abgeschaltet, um ein Einbrennen des Bildes in die Phosphorschicht des Bildschirms zu vermeiden. Dieses Zeitintervall kann mittels Tastendruck um jeweils zwei Minuten heraufgesetzt werden.

☐ Disk Copy Window: Dies entspricht im wesentlichen dem normalen Diskettenkopiervorgang, so wie er auch von der Workbench ausgeführt wird. Hier sieht man jedoch ein Fenster mit Icons zum Anklicken, braucht also nichts einzutippen.

☐ Zing! Task Monitor: Dies ist eine Zugabe für alle diejenigen, die sich für das Multitasking des Amiga interessieren. In einem Fenster werden dabei alle Aufgaben (Tasks), die der Computer gerade ausführt, angezeigt. Sehr gut sieht man auch die Prioritäten der einzelnen Programmteile, nach denen der Amiga beim Abarbeiten vorgeht.

Suchen wie ein Weltmeister

☐ Format Window: Entsprechend dem Diskcopy-Window wird hier ein Formatieren der angeschlossenen Speichergeräte mit Namensbezeichnung ermöglicht.

☐ Set Zing! Defaults. Die Voreinstellungen, mit denen das gesamte Zing!-System geladen wird, können vom Benutzer verändert werden. Darin ist die Funktionstastenbelegung im File System Window, das Schreiben von Hilfsfiles und das Suchsystem für Filenamen enthalten. Das Schreiben von speziellen Zing!-Hilfsfiles auf Disketten, Harddisks oder auch ins RAM erleichtert Zing! die Übersicht über den Inhalt der Directories. Außerdem wird

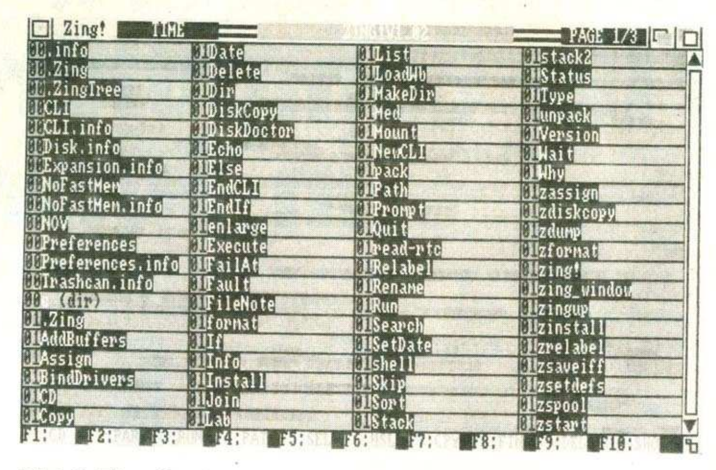


Bild 3. Eine Hardcopy des »File Tree«-Bildschirms mit vierspaltiger Ausgabe

die Diskettenarbeit dadurch wesentlich beschleunigt, da Zing! sich die Informationen nur noch aus diesen Hilfsfiles holen und nicht mehr langwierig auf der Diskette zusammensuchen muß. Das System ist eigentlich nichts besonderes, da alle ähnlichen CLI-Utilities auch solche Files erstellen. Aber nur Zing! kann dies von sich aus unterdrücken, was bei Originalen oder fast vollen Disketten zu empfehlen ist.

☐ Show this page: Zeigt den Help-Screen auf dem Bildschirm an.

Die CLI-Benutzeroberfläche

Der Hauptteil von Zing! ist das File System Window, welches jederzeit mit der linken Amiga-Taste und <F1> aufgerufen werden kann. In diesem Fenster (Bild 2) sind alle Funktionstasten nochmals einzeln belegt. Darin gelten also die ständig vorhandenen F-Tasten-Belegungen, die mit zusätzlichem Druck auf die linke Amiga-Taste erreichbar sind und darüber hinaus wird über Druck auf die einzelne Funktionstaste die Ausführung von insgesamt zehn verschiedenen Befehle möglich. Da das »File System« allerdings noch mehr als nur zehn Befehle bietet, kann man sich mit der eben schon angesprochenen Voreinstellung <left Amiga + F9> eine individuelle Belegung schaffen, die dann beim Hochfahren von Zing! immer wieder vorhanden ist. Den Großteil des Bildschirms nimmt eine Anzeige der Files der aktuellen Directory ein. Die Darstellung kann auf maximal vier Spalten erweitert werden (Bild 3). Sollten mehr Files in einer Directory enthalten sein, so werden diese auf bis zu sieben Bildschirmen (Pages) aufgeteilt. Da maximal 80 Eintragungen auf dem Bildschirm Platz haben, ergibt sich auch die Obergrenze von 500 Files oder in etwa 100 Directories mit Subdirectories, die sich von Zing! aus auf einmal verwalten lassen. Dies ist selbst für eine Festplatte mit 20 MByte noch ausreichend. Wählt man die Gesamtanzeige aller Files, so stellt Zing! vor die einzelnen Filenamen eine Zahlenmarkierung, die anzeigt, in welchem Directory sich dieses File gerade befindet (zum Beispiel 00 für das Root-Directory, 01 für das erste Subdirectory). Auf den ersten Blick ist dieses System vielleicht etwas unübersichtlich, wenn man sich jedoch einmal eingearbeitet hat, erkennt man schnell, welche Vorteile dahinter stecken. Man ist immer genau darüber informiert, wo welches File tatsächlich steht; leicht lassen sich Files von einem Directory in ein anderes verschieben; man kann Files unter bestimmten Gesichtspunkten heraussuchen, egal in welchem Directory sie gerade stecken.

Die Directories im Griff

Alle 27 Kommandos des File Systems sind über Menüs aufrufbar. Zehn davon sind noch auf zwei weitere Arten erreichbar: erstens über Funktionstasten, zweitens durch Anklicken der Icons dieser Funktionstasten am unteren Bildschirmrand. Da es unmöglich ist, alle Optionen wirklich ausschöpfend zu erklären, seien hier nur die wesentlichen Gesichtspunkte angesprochen. Das Selektieren von Files ist beispielsweise eine der wesentlichen Stärken des File System Windows. Außer den Optionen »Select all« und »Unselect all« kann man auch bestimmte Files über ein sogenanntes Pattern suchen. Dabei wird bei der Abkürzung des zu suchenden Filenamens eine »Wild-card« verwendet. Ähnlich dem schon auf dem Commodore 64 benutzten Joker wird auch hier ein Sternchen (*) eingesetzt, mit dem Unterschied, daß sich auch mehrere dieser Wildcards anbringen lassen.

Zusätzlich zur Suche mit Wortteilen darf man aber auch Zeitdaten als Auswahlkriterium mit heranziehen. Der Amiga fügt jedesmal beim Speichern eines Files das Datum, welches die Systemuhr anzeigt, mit auf Diskette. Da man beim Booten des Zing!-Systems auch nach der aktuellen Uhrzeit gefragt wird und diese wenn nötig nachstellen sollte, sind nun solche Suchbegriffe wie, »Alle Files, die ein »j« irgendwo im Namen haben und seit gestern gespeichert worden sind«, zulässig. Als Zeitbegriffe kennt das System exakte Daten, Wochentage aber auch »Today« oder »Yesterday«. Als ganz besonderer Clou darf auch noch über eine Änderung der Voreinstellungen des Programms der Suchmodus nach Wortteilen vom Wild-card-Prinzip auf das sogenannte »Advanced Pattern Matching« umgestellt werden. Dies ist nichts anderes als das Abkürzungsprinzip, welches das CLI von sich aus normalerweise anbietet (mit #? und ähnlichem). Eine weitere Eigenschaft von Zing! ergibt sich durch seinen modularen Aufbau. Programmfiles können direkt von Zing! aus mit »RUN« oder »EXECUTE« gestartet werden. Es ist außerdem ein neuer komfortabler Fileeditor und Filebrowser (ähnlich dem »TYPE«-Befehl) mit auf der Diskette. Ein echter Printer-Spooler, der nebenbei Files auf dem Drucker ausgibt, läßt sich aktivieren. Anzeigen über die aktuelle Systemzeit und den belegten Speicherplatz von Programmen auf Diskette oder im RAM ergänzen dieses hervorragende System noch. Trotz der vielen Möglichkeiten, die in Zing! stecken, wird man schnell mit dem System vertraut. Darin liegt vielleicht die einzige Falle bei der ganzen Sache. Von soviel Komfort kommt man nicht mehr los.(jk)

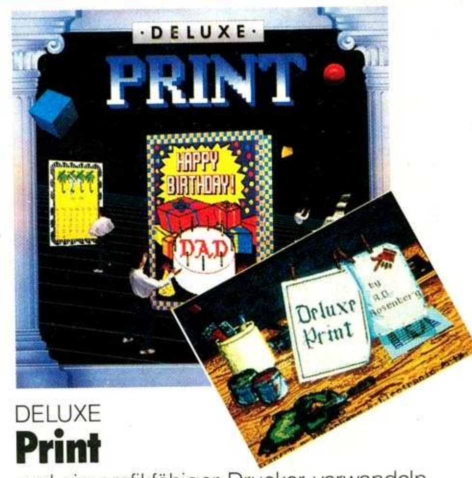


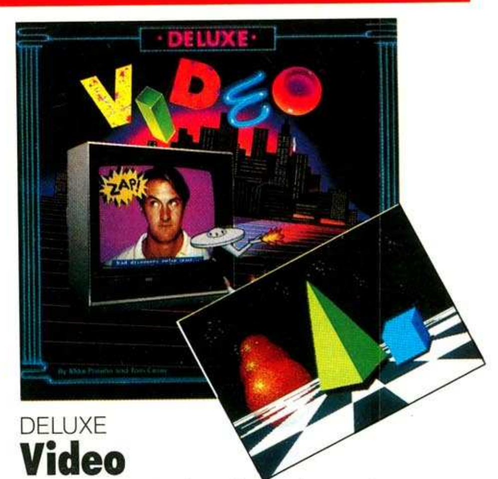
ELECTRONIC ARTS

ELECTRONIC ARTS DELUXE-Serie

Exklusivvertrieb bei Markt&Technik







NEU

DELUXE
Paint II

Deluxe Paint II ist eines der außergewöhnlichsten Grafikprogramme auf dem Softwaremarkt. Im Vergleich zu seinem Vorgänger ist Deluxe Paint II in der Lage, in der Textfunktion Fett- und Kursivschrift sowie Unterstreichungen einzubauen, perfekte Manipulation in der dritten Dimension darzustellen oder wichtige Parameter wie Längsdruck, Schwarzweißgraustufen - oder Farbdruck, Ränder und vieles mehr unmittelbar vor dem Druck per Mausklick zu verändern.
Hardware-Anforderungen:
Amiga 512 Kbyte und Farbmonitor
Bestell-Nr. 52568
DM 299,–* (sFr 249,–/öS 2990,–)
DELUXE **Paint I**
Hardware-Anforderung:
Amiga 256 Kbyte und Farbmonitor
Bestell-Nr. 52565
Nur noch DM 199,–* (sFr 199,–)

In Vorbereitung:
Music Construction Set
Das professionelle Musikprogramm für Ihren Amiga!
Hardware-Anforderung: Amiga 512 Kbyte.
Bestell-Nr. 52569
DM 249,–* (sFr 199,–/öS 2290,–*)

DELUXE
Print

und ein grafikfähiger Drucker verwandeln den Amiga in eine Druckmaschine. Sie können Karten, Poster, Briefköpfe und vieles mehr auf einfachste Weise entwerfen und ausdrucken. Besitzer eines Farbdruckers können ihr Werk auch in Farbe aufs Papier bringen. »Deluxe Print« ist kompatibel zu »Deluxe Paint«. Das bedeutet, daß man Grafiken zwischen den Programmen austauschen kann.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52566
DM 249,–* (sFr 199,–/öS 2290,–*)

DELUXE
Video

dient zum einfachen Entwerfen und Zusammenstellen von animierten Grafik-Sequenzen. Sie können so Videofilme mit Computergrafik versehen und regelrechte Computer-Videoclips zusammenstellen. Das Programm ist ebenfalls kompatibel zu »DELUXE PAINT« und »DELUXE PRINT«.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52567
DM 249,–* (sFr 199,–/öS 2290,–*)
*inkl. MwSt. Unverbindliche Preisempfehlung

Die ideale Ergänzung:
Das Buch zur Software
Markus Breuer
DELUXE Grafik mit dem Amiga
1986, 370 Seiten
ISBN 3-89090-412-2
Best.-Nr. 90412
DM 49,– (sFr 45,10/öS 382,20)
Erhältlich bei Ihrem Buchhändler

Deluxe Paint, Deluxe Print und Deluxe Video erhalten Sie in den Fachabteilungen der Kaufhäuser, in Computershops, bei Commodore-Fachhändlern oder direkt beim Verlag gegen Vorauskasse.



Zeitschriften · Bücher
Software · Schulung

703228

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

Fremdlaufwerke für den Amiga im Test

AMIGA TEST **Neben Commodore bieten auch einige andere Hersteller 3½-Zoll-Floppystationen für den Amiga an. Im folgenden Bericht erfahren Sie, ob sich diese Produkte im Alltagsbetrieb bewährt haben.**

Der Amiga wird von Commodore mit einem eingebauten 3½-Zoll-Diskettenlaufwerk ausgeliefert. Arbeitet man jedoch eine Weile mit dem Computer, so wird deutlich, daß ein einziges Laufwerk nicht immer ausreicht. Es sollte zumindest noch eine zweite, externe Floppystation vorhanden sein.

Dieses Problem haben einige Hardware-Hersteller erkannt, und sie bieten aus diesem Grund für den Amiga externe Laufwerke an. An ein solches Produkt werden natürlich gewisse Ansprüche gestellt. Es soll kompatibel zu den Original-Commodore-Laufwerken sein und nicht soviel kosten.

Wir haben uns mit sieben 3½-Zoll und einem 5¼-Zoll-Laufwerk auseinandergesetzt, wobei der Hauptteil unseres Testes darin bestand, die Diskettenstationen an den Amiga anzuschließen und unter Alltagsbedingungen am Arbeitsplatz einzusetzen. Von den äußerlichen Eindrücken einmal abgesehen, wurde schnell deutlich, daß ein ganz bestimmter Laufwerkstyp in fast allen Produkten zu finden war. Sämtliche Hersteller der Testgeräte bevorzugen 3½-Zoll-Laufwerke von NEC, wobei entweder der Typ 1035 oder 1036 A zum Einsatz kam. In der 5¼-Zoll-Diskettenstation fand ein Teac 55FV-13-U Verwendung.

Ob die Diskettenlaufwerke der Fremdhersteller zu den Commodore-Laufwerken kompatibel sind, ist kein Thema. Da der Amiga die Programme zur Bedienung der Floppystationen in seinem Betriebssystem enthält, handelt es sich bei den Diskettenlaufwerken nicht, wie zum Beispiel bei der 1541 für den C 64, um »intelligente« Peripheriegeräte. Wir haben es lediglich mit der Mechanik und einer kleinen Steuerelektronik zu tun; den Rest erledigt der Computer. Inkompatibilitäten sind aus diesem Grund praktisch ausgeschlossen: Alle Laufwerke arbeiten zweiseitig mit 80 Spuren pro Seite und erlauben eine Speicherkapazität von formatierten 880 KByte je Diskette.

Ebenfalls ausgeschlossen sind Wärmeprobleme beim Dauereinsatz. Bei unseren Testkandidaten handelt es sich ausschließlich um externe Laufwerke, die neben oder auf dem Computer stehen und die Spannungsversorgung direkt vom Amiga erhalten. Es existiert also kein heizendes Netzteil und auch keine »hitzige« Mikroprozessor-Logik. Durch die Stromzufuhr über das mehradrige Parallelkabel, mit dem die Geräte an den Computer angeschlossen sind, entfällt übrigens auch ein weiteres, häufiges Ärgernis: der »Kabelsalat«.

Die Leisetreter

Eine angenehme Überraschung haben sämtliche Diskettenlaufwerke parat, wenn sie zum ersten Mal in Betrieb genommen werden. Gemeint ist die Lautstärke. Wer das im Amiga eingebaute Laufwerk kennt, weiß um den »Lärm«, den es während der Arbeit von sich gibt. Dieser Lärm ist bei unseren Testkandidaten kein Thema: Sie schnurren allesamt leise vor sich hin und stören den gestreßten Anwender nicht im geringsten. Sogar das für den Amiga typische Klicken der Laufwerke, wenn bei geladener Workbench keine Diskette eingelegt ist, ist bei unseren Testgeräten praktisch unhörbar.

In Bild 1 sehen Sie ein 3½-Zoll-Einzellaufwerk von AB Computersysteme. In seinem Metallgehäuse enthält es eine zusätzlich abgeschirmte Mechanik vom Typ NEC 1036 A. Das Gerät kostet 388 Mark und ist damit einer der preiswerten Vertreter in unserem Test. Gefallen hat uns vor allem, daß das Laufwerk unabhängig vom Computer ausgeschaltet werden kann. Auf diese Weise erspart man sich das Abziehen der Steckverbindung zum Computer und das damit verbundene Ausschalten des Amiga. Im Betrieb zeigt das AB-Laufwerk keinerlei Schwächen; lediglich das Gehäuse hätte stabiler ausfallen können. Einen wackeligen Eindruck machte auch das angeschlossene Kabel für die Datenübertragung und die Stromversorgung, das keinen zu starken Knickbelastungen ausgesetzt werden sollte. Die nächsten Testkandidaten kom-

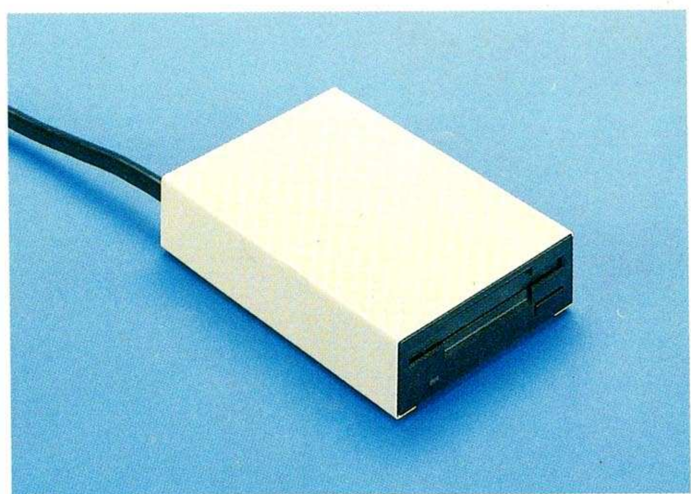


Bild 1. Handlich und klein: das 3½-Zoll-Laufwerk mit Metallgehäuse von AB Computersysteme



Bild 2. Die Amiga-Laufwerke von Compware. Die Gehäuse sind aus stabilem Kunststoff in »Amiga-Färbung«.

men von Compware und sind in Bild 2 zu sehen. Hier handelt es sich um ein Einzel- und ein Doppellaufwerk, wobei die Single-Version mit dem NEC 1035 und der »Doppeldecker« mit zwei NEC 1036 A-Laufwerken ausgerüstet ist. Die Gehäuse sind aus Plastik; das Einzellaufwerk ist auf Wunsch jedoch auch in einer Metallummantelung lieferbar. Die Preise sind 385 für das Einzel- und 650 Mark für das Doppellaufwerk; in unserem Test die preiswertesten Geräte.

Im Betrieb leisteten sich die Compware-Kandidaten ihren Kontrahenten gegenüber keine Schwächen. Auch sie arbeiteten leise und zuverlässig und demonstrierten ihre Qualität in der guten Verarbeitung. Zu Problemen kam es höchstens beim Einsatz des Einzellaufwerks in geringer Entfernung zu einem Monitor. Da es nur teilweise abgeschirmt ist, machte sich manchmal die Störstrahlung des Datensichtgeräts bemerkbar. Diese Probleme können aber durch die Anschaffung der Version mit Metallgehäuse (laut Hersteller ohne Aufpreis) vermieden werden. Wünschenswert wäre das Vorhandensein von Schaltern an jedem Laufwerk gewesen, um sie auch bei eingeschaltetem Computer und eingestecktem Kabel vom System trennen zu können.

Ein weiteres 3½-Zoll-Laufwerk kommt von Flesch & Hörnemann. Es nennt sich »Amigos« und ist mit 498 Mark das teuerste 3½-Zoll-Einzellaufwerk in unserem Test. Das Produkt besticht vor allem durch seine sehr solide und kompakte Bauweise (Metallgehäuse), die geringe Geräuschentwicklung und die Möglichkeit, mehrere Floppystationen bequem zusammenzuschalten (Bild 3). Auch hier kommt wieder ein NEC 1036 A zum Einsatz, wobei ein einzelnes Abschalten der Laufwerke leider nicht möglich ist.

Zwei weitere Vertreter der 3½-Zoll-Laufwerke für den Amiga standen uns von Soyka-Datentechnik zur Verfügung (Bild 4). Auch hier handelt es sich wieder um ein Einzel- und ein Doppellaufwerk, wobei in jedem Gerät ein, beziehungsweise zwei NEC 1036 A ihren Dienst leisten. Die Laufwerke sind in einem Plastikgehäuse untergebracht und verfügen über Schalter, um sie unabhängig aus- und wieder einzuschalten. Die Preise sind 389 Mark für das Einzel- und 689 Mark für das Doppellaufwerk. Die Geräte zeigten im Betrieb keinerlei Schwächen auf und funktionierten, wie die übrigen 3½-Zoll-Kandidaten, mit allen getesteten Programmen einwandfrei.

In einem schneeweißen Kunststoffgehäuse präsentiert sich das letzte getestete 3½-Zoll-Laufwerk von Promicro (Bild 5). Es kostet 448 Mark und beeindruckt durch eine besonders stabile Verarbeitung. Als Laufwerk findet ein NEC 1035 Verwendung, das sich von einem 1036 A vor allem durch seine größere Bauform und die nur teilweise Abschirmung unterscheidet. Als einziges Produkt in unserem Test besitzt die Promicro-Diskettenstation auf der Rückseite eine Buchse, an der ein weiteres Disketten-

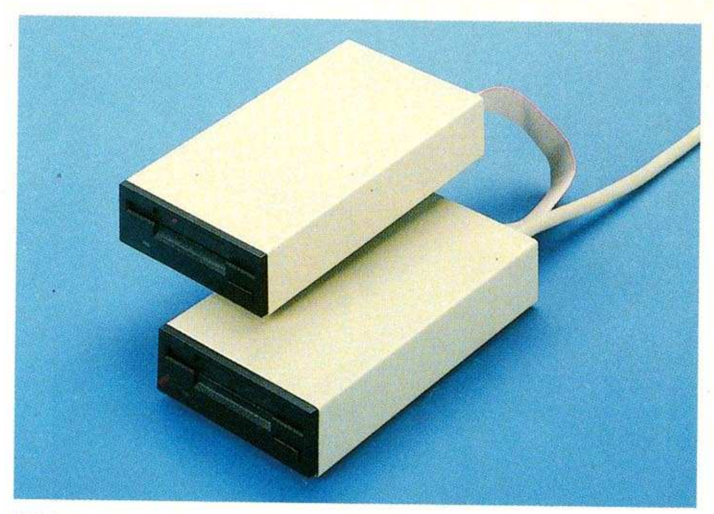


Bild 3. »Amigos« von Flesch & Hörnemann überzeugt durch solide und kompakte Bauweise

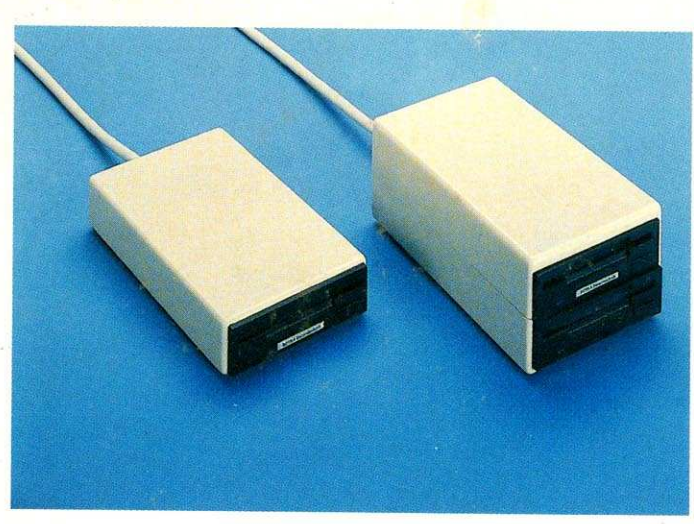


Bild 4. Die beiden Laufwerktypen von Soyka Datentechnik: einzeln abschaltbar und zuverlässig

laufwerk (auch von einem anderen Hersteller) direkt angeschlossen werden kann. Ein Öffnen des Gehäuses, um das zusätzliche Anschlußkabel unterzubringen, entfällt damit. Das Fehlen eines Schalters zum Trennen des Laufwerks vom Computersystem beeinträchtigt das positive Gesamtbild des Laufwerks nur wenig.

Neben den 3½-Zoll-Laufwerken hatten wir auch einen Kandidaten aus der 5¼-Zoll-Reihe für unseren Test zur Verfügung (Bild 6). Die Diskettenstation kommt von Compware und enthält in ihrem stabilen Metallgehäuse ein Teac 55FV-13-U-Laufwerk. Der Preis liegt mit 498 Mark über dem Durchschnitt der 3½-Zoll-Typen; scheint jedoch für das robuste und betriebssichere Laufwerk gerechtfertigt. Der Anschluß an den Computer erfolgt analog zu den 3½-Zoll-Laufwerken, und auch die Speicherkapazität ist identisch: 880 KByte formatiert. Zusätzlich ist auf der Rückseite der Floppystation ein Schalter eingebaut, der das Umschalten zwischen 40 und 80-Spur-Betrieb erlaubt. Auf diese Weise kann das Laufwerk beim Einsatz von Sidecar mit MS-DOS-Disketten arbeiten. Auch die 3½-Zoll-Diskettenlaufwerke sind dazu in der Lage; sie formatieren jedoch auch unter diesem Betriebssystem 80 Spuren je Diskettenseite und sind aus diesem Grund nicht kompatibel zu den Standardformaten von MS-DOS.

Alles in allem ein positives Bild, das wir in unserem Test von sämtlichen Kandidaten gewonnen haben. Außer ein paar »Schönheitsfehlern« sind die Produkte in ihren Leistungen kaum zu unterscheiden. Sie funktionieren mit den gängigen Programmen einwandfrei,

Fortsetzung auf Seite 117



Bild 5. Schneeweiß und sehr robust präsentiert sich das 3½-Zoll-Einzellaufwerk von Promicro

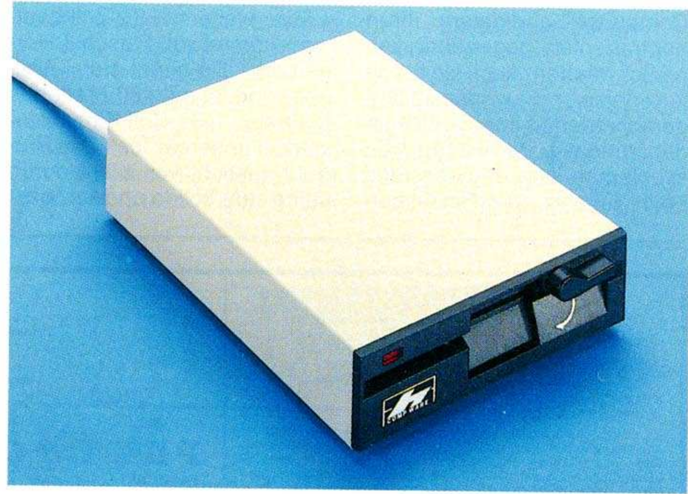


Bild 6. Ein Vertreter aus der 5¼-Zoll-Familie; umschaltbar zwischen 40 und 80 Spuren pro Diskettenseite

Hersteller	AB-Computersysteme	Compware			Flesch & Hörnemann	Promicro	Soyka	
Diskettenformat	3½ Zoll	3½ Zoll	3½ Zoll	5¼ Zoll	3½ Zoll	3½ Zoll	3½ Zoll	3½ Zoll
Anzahl Spuren/Seite	80	80	80	40/80	80	80	80	80
Laufwerktyp	NEC 1036 A	NEC 1035	NEC 1036 A	Teac 55FV-13-U	NEC 1036 A	NEC 1035	NEC 1036 A	NEC 1036 A
Anzahl der Laufwerke	1	1	2	1	1	1	1	2
Gehäuse	Metall	Plastik/Metall	Plastik	Metall	Metall	Plastik	Plastik	Plastik
Abschaltbar	ja	nein	nein	nein	nein	nein	ja	ja
Preis	388 Mark	385 Mark	650 Mark	498 Mark	498 Mark	448 Mark	389 Mark	689 Mark

Tabelle. Alle wichtigen Daten auf einen Blick

Eine Festplatte für den Commodore-Amiga

AMIGA TEST Seit einiger Zeit gibt es für den Amiga 1000 Festplattenlaufwerke auf dem Markt, die einen hohen Bedienungskomfort bei der Arbeit sicherstellen sollen. Wir haben uns drei Geräte für Sie angesehen.

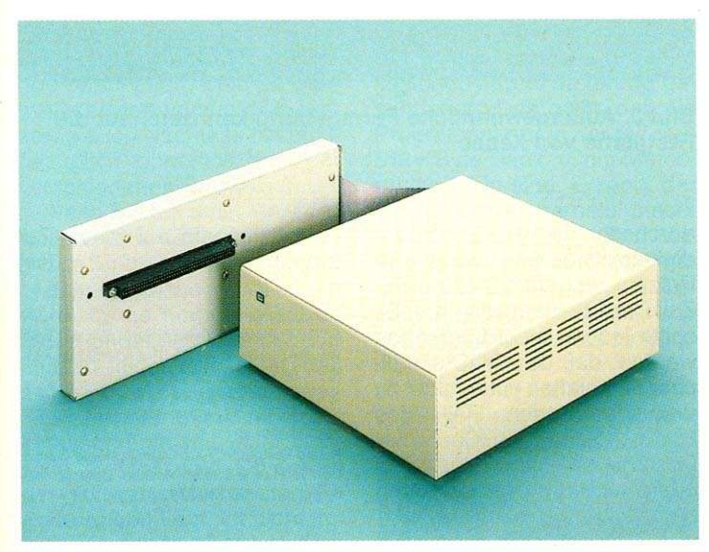


Bild 1. Schnell und leise: das Laufwerk von C Ltd mit der eingebauten 20-MByte-Rodime-Festplatte

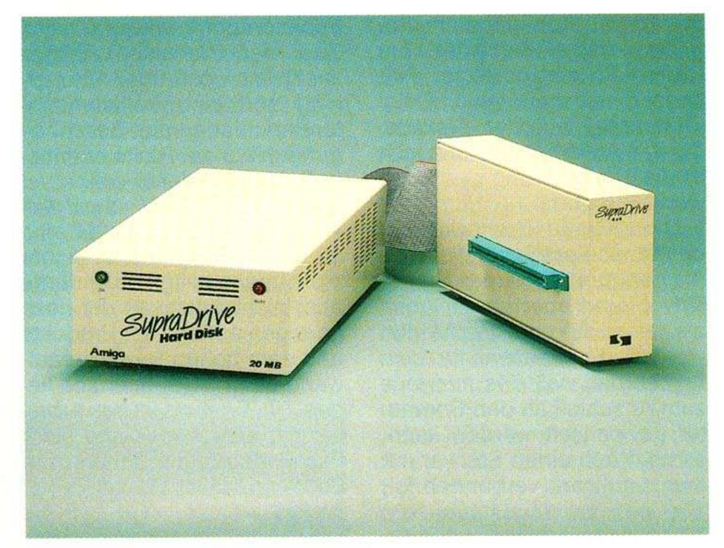


Bild 2. Besticht durch kompakte Bauweise: das »Supra-Drive« mit Mini-Scribe-Laufwerk von Computer Tuning Jürgen Kern

Bei einem Computer, mit dem die Arbeit erst ab einer Speicherkapazität von 1 MByte RAM so richtig Spaß macht, kommen Diskettenlaufwerke ziemlich schnell in Atemnot. Der Amiga betreibt die Laufwerke zum Beispiel mit einer Kapazität von 880 KByte pro Diskette, und das ist angesichts seiner Hauptspeichergröße von bis über 8 MByte nicht viel. Hier bietet sich ein Markt für Festplattenlaufwerke an, die nicht nur durch ihre Geschwindigkeit, sondern auch durch ihre hohen Speicherkapazitäten von 20 MByte und mehr bestechen.

Wir wollen Ihnen drei 20-MByte-Festplattenlaufwerke vorstellen, die alle über ein Interface am Systembus des Amiga 1000 angeschlossen werden. Versionen für den Amiga 500 sollen laut Hersteller folgen.

Das erste Produkt kommt von der amerikanischen Firma C Ltd und wird in Deutschland unter anderem von Compustore vertrieben. Hierbei handelt es sich um eine 20-MByte-3½-Zoll-Festplatte (Bild 1), die zusammen mit dem Netzteil in einem quadratischen Gehäuse untergebracht ist. Über ein mehradriges Flachbandkabel steht das Laufwerk mit dem Interface in Verbindung, das im Expansions-Port des Amiga Platz findet. Das Gerät verfügt über eine Rodime-Festplatte mit »Autoparking«, was bedeutet, daß sich das Laufwerk nach dem Ausschalten der Spannungsversorgung automatisch verriegelt und so gegen Erschütterungen, wie sie bei einem eventuellen Transport auftreten, geschützt ist.

Alle aus Amerika...

Von allen drei Festplatten in unserem kleinen Vergleichstest war die C Ltd-Version das preiswerteste und leiseste Laufwerk. Das Gerät kostet inklusive Handbuch und zwei Disketten 2345 Mark und gibt während des Betriebs nur ein leises Schnurren von sich. Mit bisher vorhandenen Programmen, sprich Datenbanken oder Malprogrammen, arbeitete die Festplatte einwandfrei zusammen; lediglich Software mit Autostart muß auf den Massenspeicher verzichten, da hier keine Möglichkeit besteht, das Betriebssystem des Amiga um die wichtigen DOS-Routinen der Festplatte zu erweitern.

Laut Compustore soll in Kürze eine neue Version der Festplatte verfügbar sein, die unter anderem mit einem deutschen Handbuch und einem um 50 Prozent schnelleren DOS ausgeliefert werden soll.

Das zweite Testmuster wurde uns von Computer Tuning Jürgen Kern zur Verfügung gestellt und nennt sich Supra-Drive. Wie Sie schon auf dem Foto (Bild 2) sehen können, handelt es sich hierbei um das kleinste der drei Laufwerke, wobei es zusammen mit dem Interface nicht größer als die C Ltd-Festplatte ist. Auch dieser Massenspeicher stand uns in der Version mit 20 MByte Kapazität zur Verfügung, wobei das eingebaute 3½-Zoll-Mini-Scribe-Laufwerk als einziges unter den drei Testkandidaten über kein »Autoparking« verfügt. Um die Festplatte dennoch transportsicher zu machen, wird ein spezielles Park-Programm mitgeliefert, das das Laufwerk anhält und bis zum erneuten Einschalten der Spannungsversorgung verriegelt.

Die Lautstärke des Supra-Drive war beeindruckend und ließ die gewohnte Ruhe am Arbeitsplatz vermissen. Betrachtet man die Geschwindigkeit, so fällt auf, daß sie ein wenig geringer als die der C Ltd-Festplatte ist, was sich im Alltagsbetrieb jedoch kaum bemerkbar macht. Der Preis des Supra-Drive liegt bei 2495 Mark, wobei neben dem Laufwerk noch eine Boot-Diskette und eine Anleitung mitgeliefert wird. Was Autostart-Programme angeht, gilt für das Laufwerk das gleiche, wie schon für die C Ltd-Festplatte: Da sie nicht von einer Workbench aus gestartet werden können, muß auf den Betrieb einer Harddisk verzichtet werden, da das Gerät nicht

konfiguriert werden kann. Alle anderen Programme liefen im Test jedoch einwandfrei.

Die dritte Festplatte in unserer Aufstellung kommt aus dem Hause Xebec (Bild 3). Hier handelt es sich um ein Gerät, das entfernt an einen Heizlüfter erinnert und von allen getesteten Massenspeichern mit Abstand das größte Gewicht aufweist. Es besitzt ein 5¼-Zoll-Laufwerk mit einer Speicherkapazität von 20 MByte und »Autoparking«. Der Preis für dieses Produkt liegt bei 2724 Mark, wobei neben der Festplatte noch eine Diskette mit dem Konfigurationsprogramm mitgeliefert wird. Ein Handbuch gibt es nicht; statt dessen existiert eine Datei »Read.Me« auf der Diskette, die den Anwender auf englisch in die Handhabung des Laufwerks einführt.

Die Geräuschentwicklung der Xebec-Festplatte ist ein wenig geringer als die des Supra-Drive, kann aber immer noch als leicht störend empfunden werden. Positiv machte sich bemerkbar, daß das Interface zum Anschluß an den Computer gewechselt werden kann, da es durch einen Stecker mit dem Laufwerk verbunden ist. Sie können das Gerät aus diesem Grund bei einem Wechsel auf ein anderes Computersystem übernehmen.

Es fällt auf, wie ähnlich sich alle drei Produkte in Wirklichkeit sind, wenn auch ihr Äußeres andere Schlußfolgerungen zuläßt. Alle Festplatten müssen »von Hand« konfiguriert werden; arbeiten also nicht mit Autostart-Programmen auf Disketten zusammen. Sind die Geräte länger in Betrieb, dann macht sich auch eine ziemliche Hitzeentwicklung bemerkbar, die im Test jedoch keinen Einfluß auf die einwandfreie Funktion der Laufwerke hatte. Zwei der Produkte verfügen zusätzlich noch über Autoparking und eine Selbstreparatureinrichtung, die Beschädigungen auf der Platte automatisch kompensieren soll.

Auffallend ist der hohe Preis. Weit über 2000 Mark für eine 20-MByte-Festplatte ist sicher nicht mehr zeitgemäß; dürfte aber nicht zuletzt an der noch geringen Auflage der Produkte und dem hohen Hardwareaufwand beim SCSI-Interface liegen. Ein Erlebnis ist das Arbeiten mit dem Amiga und einer Festplatte allemal; kommt man doch in den Genuß einer durchschnittlichen Geschwindigkeitssteigerung von etwa 30 Prozent gegenüber den 3½-Zoll-Diskettenlaufwerken. Es sollte jedoch nicht übersehen werden, daß das DOS sämtlicher Festplatten einen nicht zu unterschätzenden Teil des Hauptspeichers im Computer in Anspruch nimmt. Eine Speichererweiterung für den Amiga ist deshalb auf jeden Fall empfehlenswert. (ks)

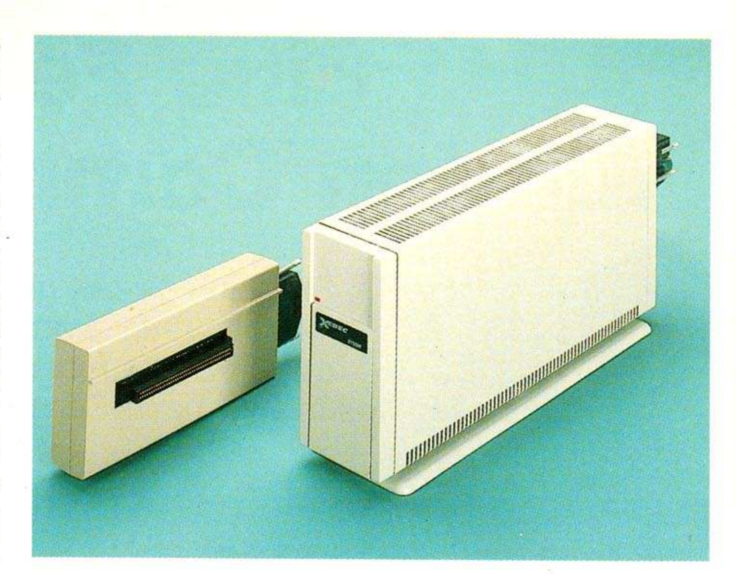


Bild 3. Außergewöhnliche Formgebung kennzeichnet die Festplatte von Xebec

Bezugsquellen:
C Ltd-Festplatte: Compustore GmbH, Fritz-Reuter-Straße 6, 6000 Frankfurt/Main, Tel. (069) 567399
Supra-Drive: Computer Tuning Jürgen Kern, Hauptstraße 95, 6074 Rödermark 3, Tel. (06074) 95142
Xebec-Festplatte: Akro Datensysteme GmbH, Carl-von-Linde-Straße 30, 8044 Unterschleißheim, Tel. (089) 3102063

RAM-Erweiterungen am Amiga

AMIGA TEST

Kauft man einen Amiga 1000, so ist er mit 512 KByte RAM ausgestattet. Für viele Anwendungen ist aber mehr Speicherplatz nützlich, teilweise sogar zwingend erforderlich. Was für den Amiga 1000 von den RAM-Erweiterungen tatsächlich zu halten ist, zeigt dieser Artikel.

Neben Floppy-Laufwerken oder Festplatten gehören RAM-Erweiterungen zu den Peripheriegeräten, die bei Amiga-Besitzern ganz oben auf dem Wunschzettel stehen. Das gewisse Mehr an Speicher ergibt nicht nur beim Programmieren, sondern auch bei vielen professionellen Programmen ein Plus an Komfort, den man bald nicht mehr missen möchte. Wir haben daher für Sie drei RAM-Erweiterungen der unterschiedlichsten Bauweise und Größe unter die Lupe genommen.

Wenn man häufig mit dem CLI arbeitet, kann dies durch eine RAM-Erweiterung zum Beispiel wesentlich beschleunigt werden. Alle Befehle, die man benötigt, werden einfach ins RAM kopiert und dem Betriebssystem wird mitgeteilt, daß es ab jetzt zum Ausführen dieser Befehle nicht mehr auf Diskette nachschauen muß. Dieser als »Assign« benannte Vorgang bringt unwahrscheinliche Geschwindigkeitsvorteile gegenüber dem normalen Arbeiten mit dem CLI, da die Programme der auszuführenden Befehle ja direkt aus dem RAM geladen werden. Auch professionelle Software, wie etwa das Mal- und Zeichenprogramm Deluxe-Paint, erblüht unter einer RAM-Erweiterung zu neuem Glanz. Wichtig ist zum Beispiel, daß man auch im Hires-Modus ganz, wie man es von Deluxe-Paint in der niedrigsten Auflösung gewohnt ist, zwischen zwei Bildschirmen zum Zeichnen hin- und herschalten kann. So lassen sich alle Möglichkeiten, die Deluxe-Paint bietet, in der höchsten Auflösung voll ausnutzen.

Alles was beim Amiga über die 512-KByte-Grenze hinausgeht, wird als »Fast Memory« bezeichnet. Der 68000-Prozessor des Amiga kann direkt auf diesen Speicherbereich zugreifen. Die Custom-Chips wie Paula oder Denise können nämlich nur 512 KByte adressieren und müssen daher immer auf den unteren Adreßbereich zugreifen.

Da sie aber bei Zugriffen über den Datenbus Vorrang gegenüber dem Hauptprozessor haben, muß dieser gelegentlich auf sie warten. Dies tritt, wie gesagt, im Speicherbereich von 512 KByte bis maximal 8 MByte nicht auf, da hier ausschließlich der Prozessor zugreifen kann.

Die hier angesprochenen RAM-Cards haben eines gemeinsam: Sie sind alle an den herausgeführten Expansions-Port des Amiga angeschlossen. Unterschiede ergeben sich jedoch schon in der Gestaltung. Entweder sind sie als flache Karten ausgeführt, wie die Microbotics-Erweiterung (Bild 1), als Kasten, auf den man noch eine Floppy stellen kann, wie der Xpander (Bild 2), oder als große Extra-Einheit, die noch durch den Mouse-Port gestützt werden muß, wie die RAM-Erweiterung von Flesch & Hörnemann (Bild 3). Letztendlich muß sich der einzelne Anwender darüber Klarheit verschaffen, welcher Bauweise er den Vorzug gibt. Vor- sowie auch Nachteile hat diesbezüglich sicherlich jede der verschiedenen Bauweisen.

Wenn Sie mit einer RAM-Erweiterung liebäugeln, gibt es einige Dinge, die Sie beim Kauf beachten sollten. Zum einen wäre da die Kompatibilität zu bestehender Software. Wir mußten leider feststellen, daß alle zur Zeit existierenden RAM-Erweiterungen nicht uneingeschränkt mit jeder Software lauffähig sind. Ob das an der Software oder an den Erweiterungen liegt, ist nicht immer eindeutig festzustellen. Daher sollten Sie sich Gedanken darüber machen, für welche Software Sie die Erweiterung einsetzen wollen und diese vor dem Kauf beim Händler mit der Erweiterung testen. Das gleiche gilt übrigens auch für zusätzliche Hardware wie zum Beispiel Festplatten. Sollen sie an den eventuell vorhandenen durchgeschleiften Systembus angeschlossen werden, könnten Sie sonst eine böse Überraschung erleben.

Prinzipiell gibt es zwei verschiedene Typen von RAM-Erweiterungen. Die einen erkennt der Amiga 1000 automatisch (autokonfigurierend). Bei den anderen muß man ihm mit Hilfe der zur Erweiterung gehörenden Software mitteilen, daß ihm nun 1 oder noch mehr MByte RAM zur Verfügung steht. Beide Erweiterungen haben ihre Vor- und Nachteile. So arbeiten zum Beispiel bei den nicht autokonfigurierenden mehr Programme fehlerfrei als bei denjenigen, die der Amiga automatisch erkennt. Der Grund ist einfach: Viel der zur Zeit existierende Software wurde für den Amiga 1000 ohne jegliche RAM-Erweiterung geschrieben. Das heißt, daß die Programme nur in der unteren 512-KByte-Bank (Chip-Memory) lauffähig sind. Erkennt der Amiga RAM oberhalb dieser 512-KByte-Grenze (Fast-Memory), so wird das Programm normalerweise in diesen Bereich geladen, was hin und wieder zum Absturz des Computers führt. Sollte das einmal der Fall sein, ein kleiner Tip: Auf der Workbench befindet sich ein Programm mit dem Namen »No Fast Mem«. Startet man dieses Programm, so wird das gesamte RAM oberhalb der 512-KByte-Grenze abgeschaltet. Bei den nicht autokonfigurierenden Erweiterungen ist dieses Programm natürlich überflüssig, denn bei ihnen muß ja das Fast-Memory erst angemeldet werden.

Die meiste kommerzielle Software arbeitete tadellos mit den Erweiterungen zusammen. Es soll jedoch nicht verschwiegen werden, daß einige wenige Programme, die speziell für den Amiga 1000 mit 256 beziehungsweise 512 KByte RAM geschrieben wurden, nicht funktionierten. Trotzdem ist jedem, der häufig mit großen Datenmengen arbeitet, zu empfehlen, sich eine solche anzuschaffen. Vergessen Sie aber nicht vor dem Kauf die Software, die Sie mit der RAM-Erweiterung einsetzen wollen, ausführlich zu testen. Kein Händler wird wohl etwas dagegen haben, denn die Erweiterungen sind nicht gerade billig. Immerhin muß für die preiswerteste über 1000 Mark bezahlt werden. Fast alle zur Zeit erhältlichen externen RAM-Erweiterungen sind in unterschiedlichen Ausbaustufen zu haben. So kostet die preiswerteste Microbotics-Erweiterung mit 512 KByte RAM 1252 Mark, mit 1 MByte 1410 und mit 2 MByte 1919 Mark.

Bis zu 8 MByte läßt sich die Erweiterung von Flesch & Hörnemann aufrüsten. Sie ist in voller Ausbaustufe natürlich mit 4538 Mark mit Abstand die teuerste. Für den, dem das zuviel ist, existieren noch preiswertere Ausbaustufen. Wer zum Beispiel die RAM-Bausteine billiger bekommt, kann für 1295 Mark die Erweiterung unbestückt beziehen. Sie soll sich problemlos bis 4 MByte aufrüsten lassen. Wer noch mehr Speicher braucht, für den existiert noch eine nachträglich einzubauende Platine, die den Ausbau bis auf 8 MByte gestattet.

Die Erweiterung, die in der Redaktion zu bestehender Software am kompatibelsten war, und sogar mit den verschiedensten Festplatten tadellos funktionierte, war der Xpander, eine 2-MByte-Erweiterung von Softwareland. Sie ist mit 1599 Mark auch nicht zu teuer. Allerdings läßt sie sich zur Zeit weder auf- noch nachrüsten. Der Hersteller plant aber auch hier unterschiedliche Ausbaustufen bis zu 4 MByte; auch an eine eingebaute Uhr ist gedacht. (jk/ah)

Bezugsquelle: Flesch & Hörnemann, Cäcilienhof 3, 4650 Gelsenkirchen, Tel. (0209) 777886
Soltau Electronic, Esplanade 39, 2000 Hamburg 36, Tel. (040) 340445
Softwareland, Franklinstr. 27, CH-8050 Zürich



Bild 1. Eine 1-MByte-Karte von Microbotics, die deutlich sichtbar einen durchgeschleiften Systembus besitzt

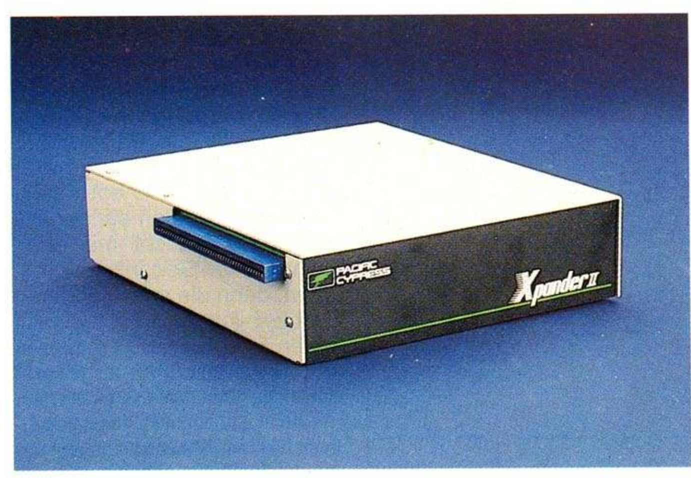


Bild 2. Auf die 2-MByte-Erweiterung Xpander kann bei Bedarf noch eine zweite Floppy gestellt werden

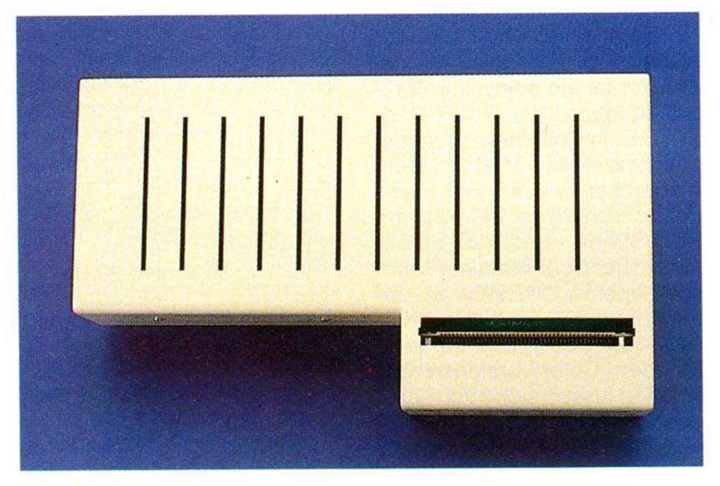


Bild 3. Sehr platzeinnehmend ist diese RAM-Card, dafür kann sie jedoch bis auf 8 MByte aufgerüstet werden

Drucker am Amiga

Der Amiga zaubert nicht nur auf dem Monitor, sondern auch auf Druckern. Wie man Drucker am Amiga 500, 1000 und 2000 anschließt und welche Drucker geeignet sind, zeigt unser Bericht.

Einen wesentlichen Teil der faszinierenden Ausstrahlung, die vom Amiga ausgeht, beruht auf seinen grafischen Fähigkeiten und der enormen Flexibilität. Will man einen Drucker an den Amiga anschließen, sollte man darauf achten, daß auch der Drucker über eine ähnliche Flexibilität verfügt wie der Computer. Zwar kann man prinzipiell fast jeden Drucker an den Amiga anschließen, da sowohl eine parallele als auch eine serielle Schnittstelle vorhanden sind, aber nicht jeder Drucker wird den Anforderungen des Amigas gerecht. Prüfen wir zunächst die verschiedenen Drucktechnologien auf ihre Eignung als Amiga-Drukker.

Seit langem bekannt sind die Typenraddrucker, die prinzipiell wie eine Schreibmaschine arbeiten und auch ein entsprechendes Druckbild haben. Sie besitzen ein hervorragendes Schriftbild, können aber nur die Zeichen darstellen, die das Typenrad enthält. Somit ist kein Grafikausdruck möglich. Typenraddrucker eignen sich also nur für den Textdruck und werden dem Amiga kaum gerecht.

Das richtige Druckprinzip

Nadel-Matrixdrucker arbeiten mit feinen Nadeln, die beim Drucken die Zeichen softwaregesteuert beliebig konfigurieren. Sie können sowohl Grafik wie auch Schrift verarbeiten. Die Druckqualität hängt aber sehr stark von der Qualität des Druckers und der Anzahl der Nadeln ab. Ein guter Matrixdrucker ist ein sinnvoller Kompromiß zwischen Grafikfähigkeit, Schriftqualität, Befehlssatz und Preis. Moderne Matrixdrucker, wie wir Sie Ihnen später vorstellen werden, mit NLQ-Schrift und LQ-Schrift sind in der Schriftqualität kaum von Typenraddruckern zu unterscheiden, erweisen sich aber als wesentlich flexibler. Mit den Color-Versionen der Drucker lassen sich auch Farbausdrucke machen. Damit eignen sich Nadel-Matrixdrucker ausgezeichnet zum Anschluß an den Amiga und werden seinen Fähigkeiten gerecht. Der wesentlichste Nachteil der Nadel-Matrixdrucker ist das relativ laute Arbeitsgeräusch.

Eine andere Art der Matrixdrucker sind die Tintenstrahldrucker. Bei ihnen werden, wie der Name schon sagt, die Schriftzeichen und die Grafik durch feine Tintendüsen erzeugt, die die Tinte auf das Papier spritzen. Heutige Tintenstrahldrucker sind technisch so ausgereift, daß das Schriftbild mit dem der Nadel-Matrixdrucker vergleichbar ist. Der Vorteil der Tintenstrahldrucker liegt darin, daß sie enorm leise drucken, auf Wunsch sogar in Farbe. Der Nachteil dieser Technik liegt darin, daß keine Durchschläge angefertigt werden können. Damit sind die Tintenstrahldrucker fast ideal zum Anschluß an den Amiga geeignet, denn sie haben alle Vorteile eines Nadel-Matrixdruckers, ohne dessen Nachteile zu besitzen.

Bild 1. Auch Laserdrucker lassen sich problemlos anschließen

Pin Nummer Centron.	Signalbedeutung	Pinbelegung am Amiga 500/2000	Pinbelegung am Amiga 1000
1	STROBE	1	1
2	DATA	2	2
3	DATA1	3	3
4	DATA2	4	4
5	DATA3	5	5
6	DATA4	6	6
7	DATA5	7	7
8	DATA6	8	8
9	DATA7	9	
10	ACKNLG	10	10
11	BUSY	11	11
12	PAPER OUT	12	12
13	SELECT	13	13
14	AUTO FEED XT	—	—
15	NC	—	—
16	0V	—	—
17	CHASSIS GND	—	—
18	NC	—	—
19-30	LOGIC GND	19-25	14-22
31	INIT	16	25

Tabelle 1. Anschlußschema für Druckerkabel

Ebenfalls zu den Matrixdruckern gehören die Thermo-Transfer-Drucker. Bei diesem Oldtimer werden mit punktförmigen Elektroden Buchstaben in ein hitzeempfindliches Papier gebrannt oder von einem Spezial-Farbband auf normales Papier übertragen. Das Farbband setzt sich aus einem Trägerfilm und einer schmelzbaren Farbschicht zusammen. Durch die Erhitzung des Thermodruckkopfes schmilzt die Wachsschicht und dringt in das Papier ein. Da die Hitzeelemente abkühlen müssen, bevor der Kopf für den Druck des nächsten Zeichens bereit ist, sind der Geschwindigkeit Grenzen gesetzt. Andererseits wird durch die hohe Auflösung ein exzellentes Schriftbild erzeugt, das sich mit der Schriftqualität von Typenraddruckern vergleichen läßt. Der Vorteil dieses Drucksystems liegt in der geringen Lautstärke, der Nachteil liegt darin, daß die Druckkosten relativ hoch sind. Vor einer Entscheidung zugunsten dieses Drucksystems sollte man allerdings berücksichtigen, daß es bisher kaum mehr als eine Außenseiterrolle gespielt hat.

Laserdrucker am Amiga

Laserdrucker sind sehr flexibel anwendbar, sowohl für Text, als auch für Schwarzweiß-Grafik. Da die meisten Laserdrucker mit einer Centronics-Schnittstelle ausgerüstet sind, lassen sich auch Laserdrucker (Bild 1) problemlos am Amiga anschließen. Auch bei der Ansteuerung braucht man sich keine großen Gedanken zu machen, denn die meisten Laserdrucker können zum Beispiel einen Epson FX-80 emulieren und wie ein solcher angesprochen werden. Wer Grafiken auf einem Laserdrucker ausgeben möchte, muß allerdings auf eine Kleinigkeit achten. Da die Laserdrucker immer das komplette Bild in ihrem Speicher

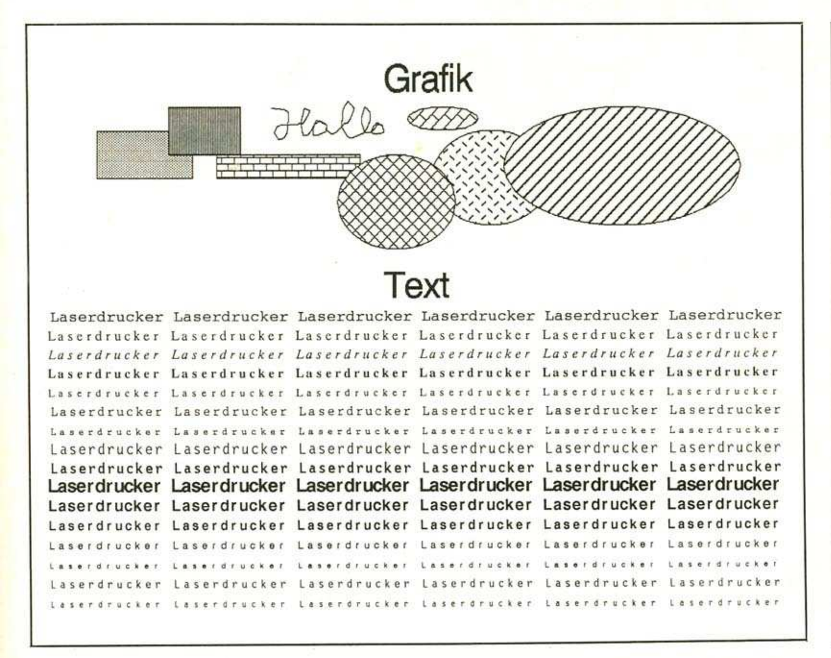


Bild 2. Qualitätsbeispiel eines Laser-Ausdrucks; in diesem Fall mit dem Sperry Modell 37 (Kyocera F1010) ausgedruckt

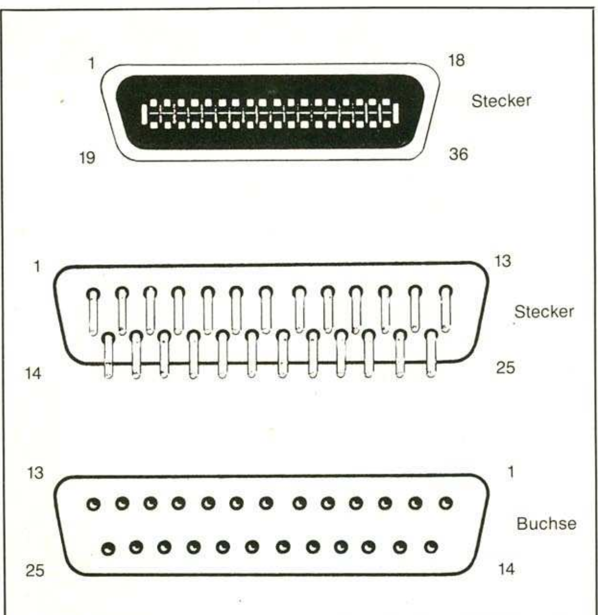


Bild 3. Diese Stecker und Buchsen brauchen Sie für das Drucker-Anschlußkabel

aufbauen, bevor es ausgedruckt wird (deshalb nennt man sie auch Seitendrucker), benötigt man einen Laserdrucker mit mindestens einem Megabyte Hauptspeicher. Diesen Speicher stellen aber nicht alle Laserdrucker zur Verfügung, sie sind deshalb im wesentlichen reine Textdrucker. Ein Beispiel mit welcher Qualität solch ein Text ausgedruckt wird, sehen Sie in Bild 2. Die Lasertechnologie hat aber nicht nur Vorteile. So sind Laserdrucker sowohl in der Anschaffung als auch im Unterhalt extrem teuer. Da Laserdrucker mit der gleichen Technik wie ein Fotokopierer arbeiten, sind sie auch genauso wartungsintensiv. Außerdem sind Toner und Drucktrommel teure Verbrauchsmaterialien. Bedenkt man dann, daß Laserdrucker mit lauten Lüftern arbeiten, nicht farbfähig sind, und nur an sauberen Orten aufgestellt werden dürfen, verfliegt einiges von der Begeisterung zugunsten dieser Technik.

Doch kommen wir zum konkreten Anschluß eines Drukkers an den Amiga. Der Anschluß eines Druckers am Amiga ist einfach. An der Rückseite des Amiga stehen zwei Buchsen zur Verfügung.

Gute Kontakte

Eine ist mit »parallel Port« und einem stilisierten Drucker bezeichnet, die andere mit »serial Port« und einem Telefonhörer. Die parallele Schnittstelle richtet sich nach der Centronics-Norm, die serielle Schnittstelle richtet sich nach der RS232-Norm. In der Regel werden Drucker an der parallelen Schnittstelle mit einem einfachen Verbindungskabel angesteckt. Ein Interface, wie es zum Beispiel beim C 64 notwendig war, kann vollkommen entfallen. Das Verbindungskabel ist auf der Computerseite mit einem 25poligen männlichen (das heißt mit sichtbaren Stiften) D-Substecker (nur Amiga 500/2000) ausgestattet. Beim Amiga 1000 wird eine Buchse (das heißt ohne sichtbare Stifte) benötigt. Auf der Druckerseite befindet sich ein 36poliger Centronicsstecker (Bild 3). Beim Amiga 500 und 2000 kann man das Kabel, das für IBM- und kompatible Computer angeboten wird, verwenden. Beim Amiga 1000 ist ein Spezialkabel notwendig. Man kann sich solch ein Kabel aber auch selber bauen. Dazu finden Sie ein Verkabelungsschema in der Tabelle 1. Gehen Sie beim Lesen der Tabelle immer von links nach rechts vor, denn beim Kabel sind nicht immer alle Pins mit gleicher Nummer verbunden. Bei den Massepins (19-30) der Centronics-Schnittstelle genügt es, wenn Sie alle diese Pins miteinander verbinden und über ein einziges Kabel an die entsprechenden Pins des Computers anlöten (Achtung, bei Amiga 1000 und Amiga 500/2000 verschieden!). Voraussetzung für eine einwandfreie Funktion ist jedoch die Aktivierung eines speziellen Druckertreibers, der die für Sonderfunktionen notwendige Anpassung vornimmt. Die in der folgenden Auflistung angeführten Druckertreiber sollten bereits auf der Workbench-Diskette Version 1.2 (unter Preferences) vorhanden sein:

— Alphacom Alphapro 101
— Brother BX 15 XL
— Commodore MPS 1000
— Commodore MPS 2000
— Diablo 630
— Diablo Advantage D 25
— Diablo C 150
— Epson JX-80
— Epson (FX,- LX,- MX,- RX-Serien)
— Fujitsu DX 2100
— HP Laserjet
— HP Laserjet Plus
— Imagewriter II
— Okidata 292
— Okidata 92
— Okimate 20
— Qume Letterpro 20

Zum Einstellen des Druckertreibers müssen Sie die Workbench-Diskette einlegen, das Directory auflisten und Preferences aufrufen. Dann können Sie den entsprechenden Druckertyp auswählen. Vergessen Sie nicht, Ihre Wahl mit deaktiviertem Schreibschutz zu speichern. Ihr Drucker ist dann bei jedem Booten ohne neue Einstellung vorgewählt. Wenn Ihr Drucker unter den oben genannten Druckertypen nicht vorhanden ist, hilft Ihnen die Kompatibilitätsliste (Tabelle 2) weiter.

Kompatibilitätstabelle

Commodore MPS 1000
— alle IBM-kompatiblen Drucker
— Drucker im IBM-Modus

Commodore MPS 2000
— NEC P6
— verschiedene 24-Nadel-drucker

Epson JX-80
— Epson EX-800 für Colordrucke
— Riteman TPX-80
— verschiedene Farbdrucker (nach ESC/P)

Epson (FX,- LX,- MX,- RX-Serien)
— Star NL (Epson Interface), ND, NR
— Citizen LSP 10, 120 D
— Panasonic
— Seikosha Al-Versionen
— Alle Drucker, die nach ESC/P-Norm arbeiten
— Laserdrucker in FX-Emulation

Fujitsu DX 2100
— Epson EX-800

HP Laserjet
Laserdrucker in HP-Emulation

Okidata 292
18-Nadel-Drucker

Tabelle 2. Für diese Drucker eignen sich die Druckertreiber des Kickstart 1.2

Der Alleskönner

Nachdem wir nun die Vor- und Nachteile der verschiedenen Druckssysteme betrachtet haben, wollen wir Ihnen einige interessante Drucker vorstellen, die sehr gut zum Amiga passen.

Der Epson EX-800 (Bild 4) zeichnet sich durch hohen Bedienungskomfort, hervorragende Schriftqualität, rasantes Drucktempo, Modultechnik für Interfaces mit bis zu 128 KByte



Bild 4. Der Epson EX-800 ist sehr vielseitig

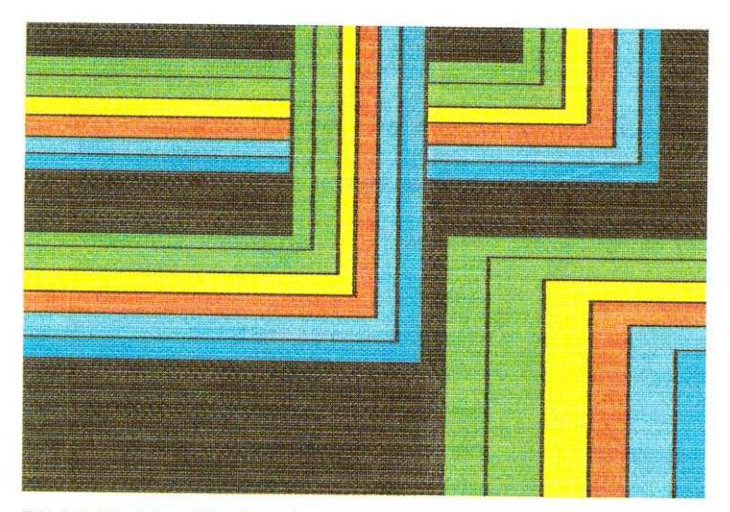


Bild 6. Farbige Hardcopien mit dem EX-800

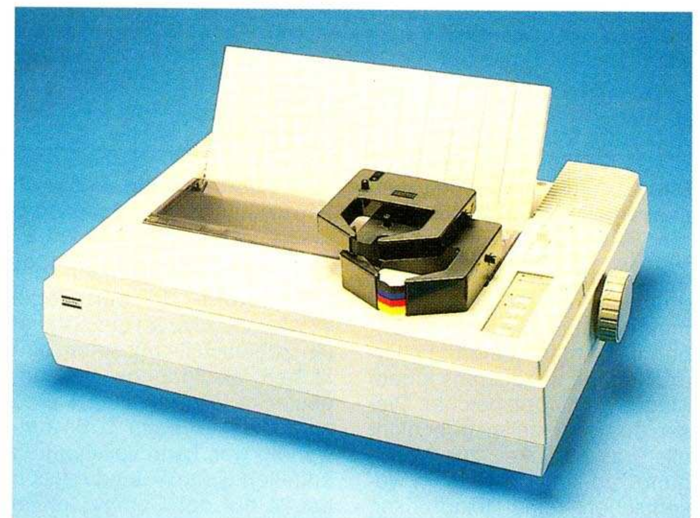


Bild 7. Seit langem bewährt, der Fujitsu DX 2100

EPSON EX-800
NLQ-Schrift1
NLQ-Schrift2
Elite-Schrift
Schmalschrift
Breit
Fettdruck
Doppeldruck
Hoch- und tief

Bild 5. Probeausdruck des EX-800

Pufferspeicher, Farbdruckfähigkeit und neuem, erweiterten Zeichensatz aus. Durch seine Fähigkeit, Papier auch rückwärts zu transportieren, kann dieser 9-Nadeldrucker auch als Ein- oder Mehrfarben-Plotter eingesetzt werden. Den Anschluß an den Computer ermöglichen ein Centronicsport und eine serielle RS232C-Schnittstelle. Der Druckspeicher ist mit 8 KByte ausreichend dimensioniert. Beim EX-800 hat sich Epson entschlossen, Endlospapier nicht mehr am Druckkopf vorbei zu ziehen, sondern zu schieben. Einzelblätter werden nach Anlegen und Ausrichten an der mitgelieferten Blattstütze automatisch eingezogen und korrekt positioniert. Die Farbbandkassette, die übrigens erstmals mit dem Druckkopf mitgeführt wird, läßt sich problemlos einlegen. Richtig Spaß macht der Umgang mit dem komfortablen Tastenfeld mit dem der EX-800 programmiert werden kann. Neben drei großen Tasten für On-/Offline, Formfeed und Linefeed sind vier LED-Anzeigen und acht Folientasten für die Wahl von zahlreichen Schrifttypen in vielen Kombinationen vorhanden. Am Bedienfeld kann man auch zwischen den beiden NLQ-Schriften des EX-800 auswählen. Das dem Testgerät beigelegte deutsche Handbuch, läßt keine Wünsche offen. Bei EDV-qualität druckt der EX-800 300 Zeichen pro Sekunde. Schönschrift (NLQ) steht in zwei Zeichensätzen zur Verfügung (Bild 5) und wird mit 50 Zeichen in der Sekunde auf das Papier gebracht. Hochauflösende Punktgrafik kann in sechs Dichten und in zwei Geschwindigkeiten für Hardcopies vom Bildschirm eingesetzt werden. Insgesamt stehen 10 Grafikmodi zur Auswahl. Gegenüber den Vorgängermodellen ist der Druck von Grafiken durch den neuen Druckmodus in zwei Richtungen fast doppelt so schnell wie bisher. Mit dem EX-800 hat Epson jetzt zum Preis von 1898 Mark einen hochwertigen und schnellen Bürodrucker mit der Fähigkeit ausgestattet, farbig zu drucken (Farbkit 250 Mark). Alle zuvor geschilderten Eigenschaften des EX können damit auch bunt gemacht werden. Für den Farbdruck wird das schwarze Druckband gegen ein Band mit den Farben Rot, Blau, Gelb und Schwarz ausgetauscht. Durch Heben und Senken des Farbbandeinsatzes wird der jeweilige Farbstreifen vor den Druckkopf gebracht und die gewünschte Farbe auf das Papier »genadelt«. Durch wiederholtes Übereinanderdrucken der vorhandenen vier Farben kann jede andere Farbschattierung erreicht werden. Farbige Bildschirmkopien (Bild 6) zu erstellen, wird mit dem EX-800 zum Vergnügen. Aber auch für die kommerzielle Grafik ergeben sich zahlreiche Anwendungsmöglichkeiten. Bunte Balken-, Säulen- oder Kreisdiagramme demonstrieren Geschäftsstatistiken augenfälliger als es ein Schwarzweiß-Bild kann. Für den Amiga mit seiner hohen Farbgrafikauflö-

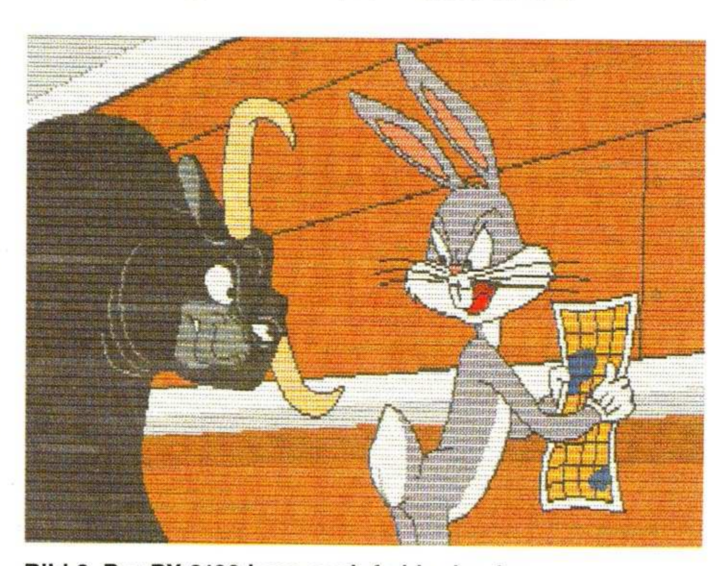


Bild 8. Der DX 2100 kann auch farbig drucken

Fujitsu DX 2100
NLQ-Schrift1
Elite-Schrift
Schmalschrift
Breit
Fettdruck
Doppeldruck
Hoch- und tief

Bild 9. Schriftprobe des DX 2100

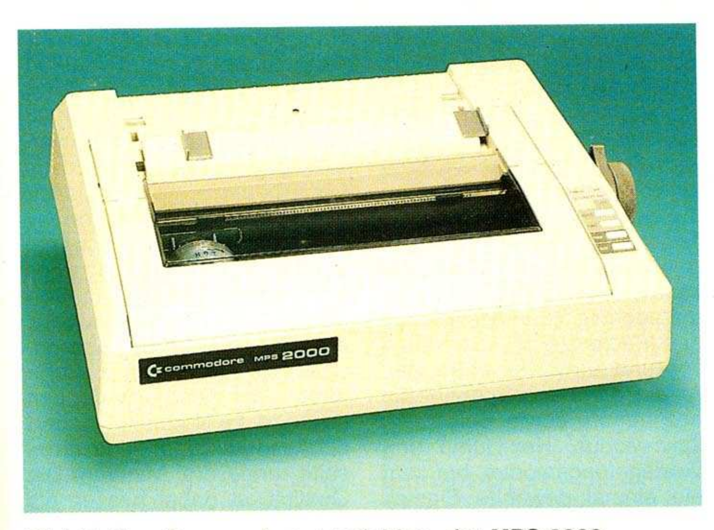


Bild 10. Von Commodore empfohlen, der MPS 2000



Bild 12. Der MPS 2000 C kann auch farbig drucken

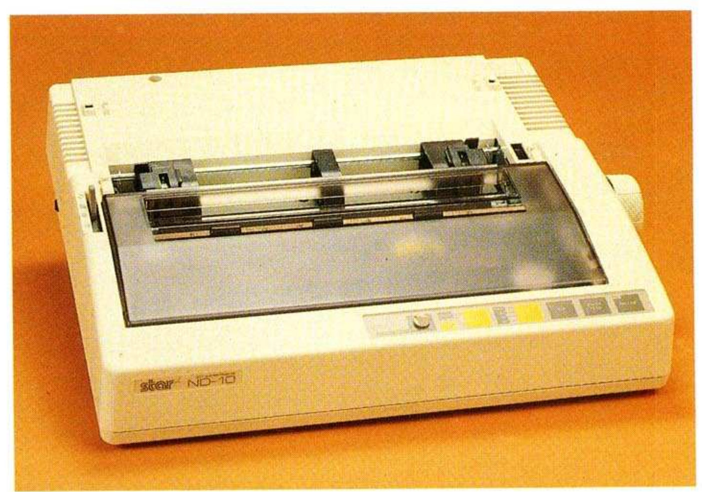


Bild 13. Preiswert und gut, der Star ND-10

Commodore MPS 2000
NLQ-Schrift
Elite-Schrift
Schmalschrift
Breit
Fettdruck
Doppeldruck
Hoch- und tief

Bild 11. Schöne Schrift mit 24 Nadeln, der MPS 2000

sung ist der EX-800 ein sehr gut geeigneter Drucker, der durch seine Qualität überzeugt.

Text und Farbe

Der Fujitsu DX 2100 (Bild 7) ist ein Drucker, der ebenfalls Schönschrift und Farbgrafik beherrscht. Er benutzt dazu, wie die meisten Drucker, einen 9-Nadel-Druckkopf. Wie beim Epson EX-800 muß man auch beim DX 2100 den Farb-Umbausatz (496 Mark) zusätzlich kaufen und einbauen. Der Einbau ist aber sehr einfach und läßt sich mit einem einfachen Schraubenzieher bewerkstelligen. Für den Amiga ist es empfehlenswert, zusätzlich zu den 1778 Mark für den Drucker auch den Farb-Umbausatz zu erstehen, denn die Qualität des Farbdrucks ist sehr gut (Bild 8). Der DX 2100 wird in einer IBM- und einer Epson-Version geliefert. Für den Amiga ist die Epson-Version besser geeignet. Die Versionen unterscheiden sich allerdings nur durch eine Steckkarte und können auch nachträglich von einer Version in die andere umgebaut werden. Der DX 2100 gehört nicht nur zu den sehr solide aufgebauten Druckern, er erreicht auch eine hohe Druckgeschwindigkeit von 220 Zeichen pro Sekunde. In der NLQ-Schrift sind es immerhin noch 44 Zeichen pro Sekunde. Die NLQ-Schrift ist sehr ansprechend und ist beinahe für jede Anwendung ausreichend (Bild 9). Das alles wäre ohne einen sinnvollen Befehlssatz aber wenig nützlich. Glücklicherweise hat man beim DX 2100 eine vollständige Auswahlpalette an ESC-Steuerbefehlen, denn sowohl die Befehle des ESC/P-Standards als auch die Farbsteuerbefehle des Epson JX-80 sind vorhanden. Wer aber lieber die Schriften mit Schaltern oder Tasten einstellen möchte, braucht beim DX 2100 nicht lange zu suchen. Die vier Funktionstasten dienen, je nachdem, in welcher Reihenfolge beziehungsweise Kombination sie gedrückt werden, dazu, 14 verschiedene Schriften oder Funktionen aufzurufen. Die Installation des DX 2100 am Amiga ist besonders einfach, denn auf der Workbench-Diskette 1.2 ist ein eigener Druckertreiber für den DX 2100 vorhanden.

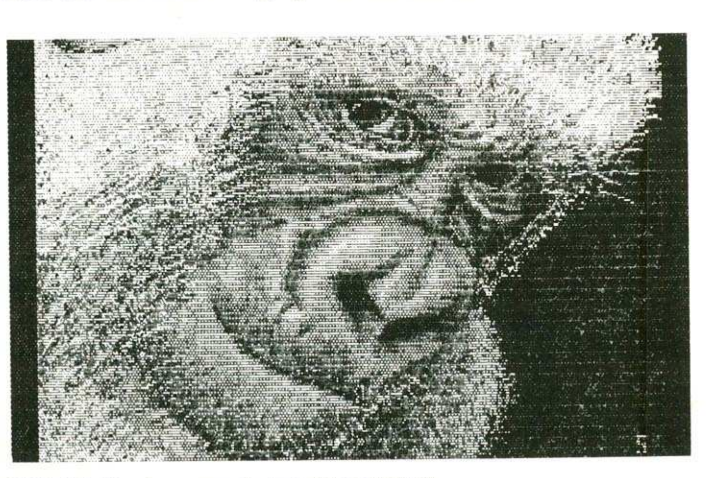


Bild 15. Hardcopy mit dem Star ND-10

Der Vielseitige

Der Commodore MPS 2000 (Bild 10) ist der Drucker, der von Commodore für den Amiga empfohlen wird. Dabei hat man die Auswahl zwischen der Schwarzweiß-Version MPS 2000 oder dem farbig druckenden MPS 2000 C. Beide Drucker sind, wie alle Drucker von Commodore, keine Eigenproduktion, sondern werden von einem Fremdhersteller geliefert. Im Falle des MPS 2000/2000 C ist das NEC mit den beiden Druckern P6 und CP6. Dieser Umstand gibt der Qualität der beiden Drucker aber keinen Abbruch, im Gegenteil, man hat den Vorteil, nicht nur bei Commodore-Händlern, sondern auch bei NEC-Händlern Service erhalten zu können. Außerdem ist der MPS 2000 mit einem Preis von 1695 Mark preiswerter als der P6. Der Farb-Umbausatz kostet 300 Mark extra, das heißt der MPS 2000 C kostet 1995 Mark. Der wesentlichste Unterschied zu anderen Druckern ist beim MPS 2000 die Anzahl der Nadeln im

Star ND-10
NLQ-Schrift
Elite-Schrift
Schmalschrift
Breit
Fettdruck
Doppeldruck
Hoch- und tief

Bild 14. Schriftprobe des Star ND-10



Bild 16. Farbig mit Tinte — der Canon PJ 1080A

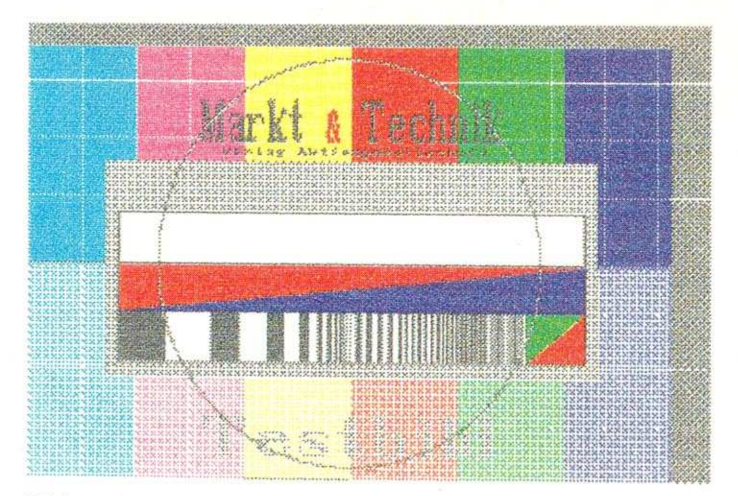


Bild 17. Leise und farbig — Hardcopy des PJ 1080A

Canon PJ-1080A

Elite-Schrift
Breit
Fettdruck

Bild 18. Nicht überzeugend ist der Textausdruck

Druckkopf. Gegenüber den sonst üblichen neun Nadeln besitzt der MPS 2000 gleich 24 Nadeln. Dadurch hat er gegenüber anderen Druckern natürlich Vorteile sowohl in der Schrift (Bild 11), als auch bei der Grafik. Buchstaben werden beispielsweise aus wesentlich mehr Punkten zusammengesetzt als bei 9-Nadel-Druckern. Auch bei der Grafik machen sich die 24 Nadeln bemerkbar. Mit dem verfügbaren bidirektionalen Traktor lassen sich sowohl einfache und schnelle 8-Nadel-Grafiken in niedriger Auflösung als auch hochauflösende 24-Nadel-Präzisionsausdrucke mit 360 x 360 Punkten pro Zoll anfertigen (Bild 12). Trotz seiner vielen Nadeln ist der MPS 2000 ein schneller Drucker, der es in Normalschrift auf 216 Zeichen pro Sekunde und in der LQ-Schrift auf sehr gute 60 Zeichen pro Sekunde bringt. Die Anpassung an den Amiga ist beim MPS 2000 natürlich sehr einfach, denn unter Preferences ist ein eigener Druckertreiber für den MPS 2000 gespeichert. Dieser Druckertreiber ermöglicht es, alle Leistungen, besonders den hochauflösenden Grafikdruck des MPS 2000, zu nutzen.

Preiswert und flexibel

Der Star ND-10 (Bild 13) ist Bestandteil einer Druckergeneration, die bereits mit dem NL-10 als kleinstes Modell letztes Jahr eingeführt wurde. Die Familienähnlichkeit zur neuen Generation ist auch dem ND-10 bereits äußerlich anzusehen. So findet man auf der Vorderseite ein umfangreiches Tastenfeld, das gesamte Gehäuse ist relativ flach und das Papier wird durch einen Schubtraktor transportiert. Das Tastenfeld besteht aus fünf Schaltern und einem Drehschalter. Mit dem Drehschalter wird die verwendete Seitenlänge in vorgegebenen Schritten in Zoll eingestellt. Alle anderen Tasten sind mit mindestens zwei Funktionen belegt. So kann man per Taste den linken und rechten Rand bestimmen, das Papier in Microschritten vor- und zurückbewegen, den Anfang der ersten Schreibzeile bestimmen, sowie Selbsttest, Italic-Modus, Hex-Dump und Verriegelungsmodus, bei dem die einmal gewählte Einstellung fixiert wird, auswählen. Außerdem kann man mit diesen Tasten NLQ- oder Draft-Modus festlegen, die Schriftbreite wählen und natürlich Seiten- und Zeilenvorschub auslösen. Der Standard-Befehlssatz des ND-10 richtet sich an der ESC/P-Norm. Dabei zeigt der ND-10 durchweg gute Leistungen. Es fiel auf, daß der ND-10 mit seinem Pufferspeicher von 12,6 KByte den Amiga nur wenig blockiert. Der ND-10 schafft in der Draft-Qualität 180 Zeichen pro Sekunde und in der NLQ-Schrift 45 Zeichen pro Sekunde. Damit ist der ND-10 ein wirklich flotter Drucker, der für einen 9-Nadel-Drucker ein sehr ansprechendes Schriftbild liefert (Bild 14) und grafikfähig ist (Bild 15). Mit diesen Leistungen und dem sehr günstigen Preis von 1295 Mark ist der ND-10 eindeutiger Favorit unter den 9-Nadel-Druckern in dieser Preisklasse. Am Amiga stellt er eine preisgünstige Alternative für alle dar, die keinen Wert auf farbigen Ausdruck legen.

Spritzig und leise

Der Canon PJ 1080A (Bild 16) ist ein Farb-Tintenstrahldrucker mit einem sehr günstigen Preis/Leistungsverhältnis. Für etwa 1498 Mark kann man an den Vorzügen der Tintenstrahltechnik teilhaben. Im wesentlichen heißt das: gleichbleibende Druckqualität von der ersten bis zur letzten Seite und die verblüffende Geräuscharmut. Insgesamt verfügt der Canon PJ 1080A über sieben verschiedene Farben. Für einen Farbkünstler wie dem Amiga wäre das natürlich viel zu wenig. Deshalb bekommt man mit dem Drucker eine spezielle Anpassungssoftware, die im Preferences-Menü installiert wird. Dieses Programm paßt das Amiga-System an den Canon PJ 1080A an, wobei die komplette Farbpalette des Amigas von 4096 Farbtönen durch Mischen verschiedener Farben und die Verwendung spezieller Muster hergestellt wird. Man kann mit dem Anpassungsprogramm auch eine Umstellung des Druckers auf Doppeldruck vornehmen. In diesem Modus kommen die Farben besonders gut zur Geltung, dafür dauert der Ausdruck doppelt so lange. Im Normalfall dauert der Ausdruck einer Bildschirmseite runde drei Minuten. Die Qualität ist dabei besser, als bei manchem Matrixdrucker, weil keine Streifen zwischen den einzelnen Druckzeilen zu sehen sind (Bild 17). Der Ausdruck bleibt auch bei mehreren Hardcopies immer gleich gut und wird nicht wie bei Druckern, die mit einem Farbband arbeiten, im Laufe der Zeit immer schlechter. Natürlich kann man mit dem Canon PJ 1080A auch Text drucken, wobei man aber diese Funktion nicht gerade als die besondere Stärke dieses Druckers bezeichnen kann (Bild 18). Für einfache Listingausdrucke reicht die Qualität immerhin aus. Bei Briefen, besonders bei wichtigen, macht sich das Fehlen der NLQ-Schrift und verschiedener Schriftvariationen doch negativ bemerkbar. Dafür ist die Handhabung dieses Druckers relativ einfach. Zum Beispiel ist der Austausch der beiden Tintenbehälter (einer schwarz und einer mit den Farben) mit zwei Handgriffen zu bewerkstelligen. Sehr oft wird dieser Austausch allerdings nicht notwendig, denn die Tintenpatronen sind sehr ergiebig. Damit ist der Canon PJ 1080A für die farbige Grafikausgabe geradezu prädestiniert. Beim Ausdruck von Texten sollte man allerdings keine zu großen Ansprüche haben. Am besten ist es, man kombiniert den Canon PJ 1080A mit einem guten einfarbigen Drucker wie dem Star ND-10. Dann hat man das beste aus zwei Welten und für jede Aufgabe einen spezialisierten Drucker. (aw)

EX-800: Epson Deutschland GmbH
Zülpicher Str. 6
4000 Düsseldorf 11
Fujitsu 2100: Fujitsu Deutschland GmbH
Rosenheimer Str. 145
8000 München 80
MPS 2000: Commodore Fachhandel
ND-10: Star Europe GmbH
Mergenthaler Allee 1-3
6236 Eschborn/Ts.
PJ 1080A: Interplan
Nymphenburger Str. 134
8000 München 19

Fortsetzung von Seite 108

sind leise bis sehr leise und machen meist einen stabileren Eindruck als die vergleichbaren Commodore-Laufwerke. Interessant ist für den Käufer vor allem der — im Vergleich zu Commodore — günstige Preis. Kompatibilitätsprobleme sind, wie schon erwähnt, nicht zu befürchten.

In der Tabelle haben wir für Sie noch einmal alle wichtigen Daten der Laufwerke zusammengefaßt. Beim Kauf sollten Sie in Betracht ziehen, für welche Aufgabengebiete Sie eines oder mehrere externe Diskettenlaufwerke benötigen. Arbeiten Sie zum Beispiel viel mit Textverarbeitung oder Datenbanken, so sollten Sie sich am besten gleich ein Doppellaufwerk zulegen. Das ist preislich günstiger, als wenn ein zweites Einzellaufwerk später nachgerüstet wird. Spielen Sie hingegen gerne, so reicht ein Einzellaufwerk aus, das aber mit einem Schalter zum Trennen vom Computersystem ausgerüstet sein sollte. Jede angeschlossene Diskettenstation verbraucht nämlich Speicherplatz im Computer, und das verträgt sich ohne Speichererweiterung mit einigen sehr aufwendigen Programmen nicht. Hier ersparen Sie sich dann das kontaktfeindliche Aus- und Einstecken des Laufwerks. Ein Metallgehäuse ist bei den Diskettenstationen nicht unbedingt erforderlich, vor allem, wenn der Laufwerkstyp NEC 1036 A eingebaut ist. Hier ist die gesamte Mechanik und auch die Steuerelektronik intern vollständig gegen Störstrahlung von außen abgeschirmt, so daß das Metallgehäuse lediglich für höhere Stabilität sorgt.

Wenn Sie sich nicht sicher sind, ob Sie zu einem späteren Zeitpunkt vielleicht ein weiteres externes Laufwerk kaufen wollen, sollten Sie darauf achten, daß eine problemlose Erweiterung möglich ist. Hier ist es von Vorteil, wenn bei der vorhandenen Diskettenstation der Anschluß für ein weiteres Laufwerk vorhanden ist. (ks)

Bezugsquellen:

AB-Computersysteme, A. Büdenbender, 5000 Köln 41, Wildenburgstraße 21, Tel. 0221/4301442

Compware, Helmut Adler Computer-Technologie, Schlägel&Eisen-Straße 9, 4352 Herten-Langenbochum, Tel. 02366/55891

Flesch & Hörnemann GbR, Cäcilienhof 3, 4650 Gelsenkirchen, Tel. 0209/777886

Promicro GmbH, Tegeler Straße 6, 1000 Berlin 65, Tel. 030/4653088

Soyka Datentechnik, Harald Soyka, Hattinger Straße 685, 4630 Bochum 5, Tel. 0234/411913

Computer-Kino

Der Amiga kann Grafiken von fantastischer Brillanz zeigen und animieren. Damit werden auch Spiele möglich, die an einen Kinofilm erinnern.

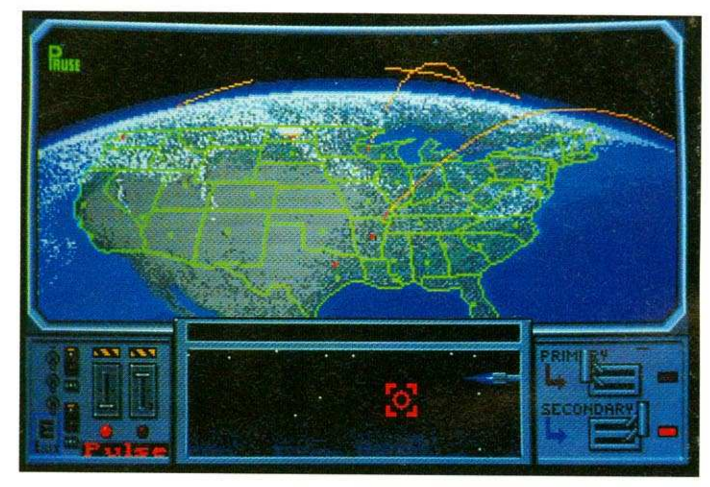
Die Spiele von Cinemaware haben einen hohen Anspruch, den man bislang nur von Infocom-Adventures kannte. Sie wollen den Spieler wie im Kino in eine neue Welt entführen, in der er aber vollkommen frei handeln kann. Er ist nicht gezwungen nur zu konsumieren, wie es im Kino der Fall ist, sondern kann selbst die Handlung entscheidend beeinflussen. Die erste Kostprobe ihres Könnens gaben die Programmierer bei »Defender of the Crown« ab. Mit »SDI« und »Sinbad« gibt es nun die nächsten Spiele.

SDI beschäftigt sich mit einem momentan hochpolitischen Thema: dem amerikanischen SDI-Programm. Im Jahr 2017 steht die Raketenabwehr im Weltraum kurz vor ihrer Vollendung. Politische Konsequenzen hatte die Errichtung nicht. Die Entspannung ist inzwischen soweit, daß es keine Konflikte gab. Statt dessen hat man im Bereich der Weltraumfahrt zusammengearbeitet und gemeinsam eine Marsexpedition durchgeführt.

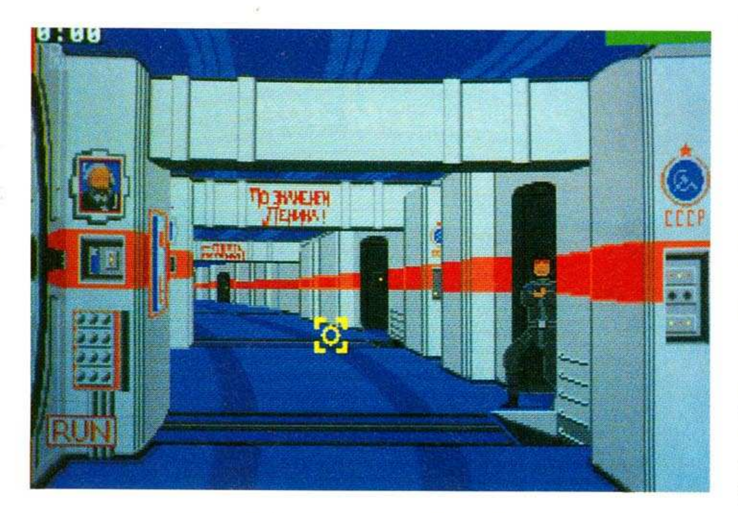
Doch am 25. Oktober 2017, am 100 Jahrestag der Oktoberrevolution, putschen der KGB und das Militär gegen die Kremlführung. Sie wollen die USA vernichten, bevor SDI vollkommen einsatzbereit ist. Der KGB befürchtet nämlich, daß SDI zu einer Abhängigkeit der UdSSR führt. Die militärischen Kreise wollen die ihrer Meinung nach letzte Chance nutzen, um den »Sieg« der USA zu verhindern. Deshalb haben Sie die wichtigsten Raketenstellungen und Raumhäfen besetzt, um die schon installierten Satelliten zu zerstören und im Erstschlag die USA auszuschalten. Das Ultimatum der Putschisten ist daher auch eindeutig: entweder wird SDI sofort gestoppt oder es kommt zum Krieg. Die offizielle Kremlführung bietet den USA in dieser Situation jede Hilfe an, um den Weltkrieg zu verhindern.

Sie übernehmen im Spiel die Rolle des Commander McCormick, der die Raumstation der USA leitet. In Ihre Hände wird die Lösung des Konflikts gelegt. Wenn Sie versagen, hat das einen atomaren Krieg zur Folge und führt zum Untergang der Menschheit.

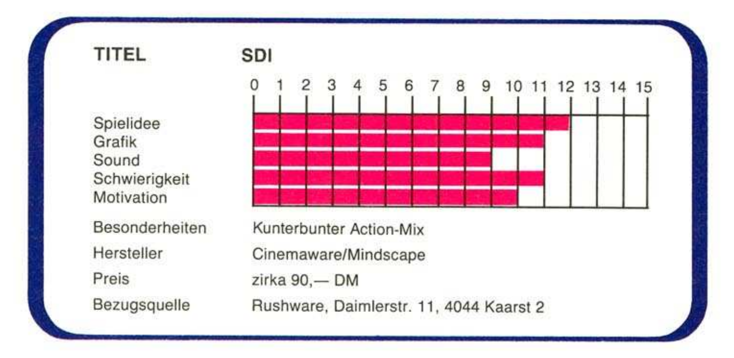
Um das zu verhindern, müssen Sie verschiedene Aufgaben lösen. Alle Aktionen werden von der großen Raumstation aus gesteuert, wo auch das Spiel beginnt. Hier gibt es große Kontrollmonitore, die zum Beispiel über anfliegende Raketen und den Zustand der Satelliten Auskunft geben. Um die Verteidigungskraft zu erhalten, heißt es die neun Laser-Satelliten funktionsfähig halten. Da diese von den sowjetischen Raumgleitern angegriffen werden, muß man die Raumschiffe abschießen und die Satelliten reparieren. Dazu steigt McCormick selbst in einen Gleiter um sich ins Schlachtgetümmel im Weltraum zu wagen. In einer 3D-Actionsequenz, die etwas an einen Flugsimulator erinnert, tritt der Spieler gegen die feindlichen Flieger an.



Raketen greifen an und müssen vernichtet werden



In der Raumstation gibt es ein packendes Duell



TITEL	SDI
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Kunterbunter Action-Mix
Hersteller	Cinemaware/Mindscape
Preis	zirka 90,— DM
Bezugsquelle	Rushware, Daimlerstr. 11, 4044 Kaarst 2

Im Gefecht kann der eigene Gleiter aber beschädigt werden und Treibstoff und Energie stehen auch nicht unbegrenzt zur Verfügung. Dann heißt es umkehren und an der Station andocken. Dieser Teil besteht aus einer schnellen Vektorgrafik, mit der die Schleuse dargestellt wird. Hier muß man sich gefühlvoll der Station nähern und die Schleuse möglichst genau in der Mitte treffen.

Nach kurzer Zeit kommen schon die ersten Atomraketen angeflogen, die von der Raumstation aus abgeschossen werden. Doch genau wie bei der Schleuse sind diese Aufgaben noch nicht von Computern und Sensoren gesteuert, sondern der Mensch muß hier mit Augenmaß und Fingerspitzengefühl arbeiten. Sobald die Raketen auftauchen, muß man sie per Hand anvisieren und abschießen. Da sehr viele Raketen aus mehreren Richtungen gleichzeitig anfliegen und nicht immer der ideale Verteidigungssatellit zur Verfügung steht, ist es sehr schwer, alle Atomsprengköpfe vor dem Einschlag zu neutralisieren.

Die letzte Aufgabe des Spiels wartet in der Raumstation der UdSSR, wo die aparte Kommandantin Natalya, die Sie übrigens bei der gemeinsamen Marsexpedition kennengelernt haben, von den Putschisten gefangengehalten wird. Dringt man in die Station ein, heißt es in einer Schlacht Mann gegen Mann KGB-Agenten mit einem gezielten Schuß aus der Laserpistole abzuschießen. Der Korridor bis zu Natalyas Räumen ist sehr lang, und McCormick kann sich immer nur einige Meter weiter vorkämpfen. Sobald er eine neue Deckung gefunden hat, müssen Sie erst wieder solange Leute töten, bis das Teilstück frei ist und Sie weiter vorrücken können. Hat man sich verschätzt, stirbt der gute McCormik den Heldentod in der Raumstation. Das Gemetzel im Gang ist zwar nicht sonderlich geschmackvoll, führt aber letztendlich in die Gemächer von Natalya. Hier haben die Programmierer sehr viel Humor bewiesen, denn alles sieht aus wie die Liebesszene bei »Defender of the Crown«. Statt des Kamins sorgen die Sterne für eine sanfte Beleuchtung mit Gegenlicht, der Held betritt den Raum, verharrt einige Sekunden und man wartet auf die holde Russin, die sich schon ein paar Mal über Funk im Raumgleiter gemeldet hat. Doch statt eines tollen Rendez-

vous in der Raumstation sieht man nur, wie McCormick einige Sekunden später mit kernigem Schritt die Raumstation verläßt.

SDI enthält viele verschiedene Spielsituationen, die das Programm sehr komplex machen. Man darf sich keine größeren Fehler erlauben, um überhaupt eine Chance zu haben, den Weltkrieg zu verhindern. Die Grafik ist gut gelungen, was besonders für die Zwischenszenen gilt, die mit dem eigentlichen Spiel nicht viel zu tun haben. Leider hapert es etwas mit der Animation, denn die Figuren bewegen sich teilweise recht ruckartig.

SDI kann man eine gute Grafik und viele Spielstationen bescheinigen. Einige Sequenzen sind aber noch nicht völlig ausgereift, so daß sie einzeln nicht überzeugen können. Alles in allem bietet SDI aber viel Spiel fürs Geld.

Nachdem SDI in den luftlosen Höhen des Erdorbits spielt, entführt Sinbad den Spieler in die Märchenwelt von 1001 Nacht. Der greise Kalif von Damaron wurde von geheimnisvollen Mächten in einen Falken verwandelt. Wenn er nicht innerhalb kürzester Zeit wieder zurückverwandelt wird, bleibt er für immer ein Falke und kann nicht seinen Nachfolger bestimmen. Doch gerade zu dieser Zeit ist es wichtig, einen starken und entschlossenen Führer zu haben, denn der Schwarze Prinz bedroht mit seinem Heer das Reich. Es gilt so schnell wie möglich das Gegenmittel zu finden, von dem niemand weiß, wo und was es ist. Vielleicht hat aber die Sage über die Statuen des Volks der Ishtiki etwas mit der Lösung zu tun? Das ist genau die richtige Aufgabe für einen Abenteurer wie Sinbad. Ihn verbindet eine tiefe Freundschaft mit der Familie des Kalifen und er ist auch in ihrem Haus aufgewachsen. Dort hat er auch die schöne Prinzessin Sylphani kennengelernt, doch bevor es zwischen den beiden ernst wurde, hat sich Sinbad lieber auf seine erste Reise gemacht und erst jetzt, zehn Jahre später, trifft er seine Jugendliebe wieder.

Sinbad muß um die gesamte Welt segeln, um irgendwo Leute zu treffen, die ihm weiterhelfen können. Während er zu Wasser oder auf dem Land durch die Gegend streift, tauchen ab und zu auch Personen auf, die sich mehr oder minder bereitwillig mit ihm unterhalten. Dazu gibt es in einem speziellen Teil ein Pull-Down-Menü, aus dem man sich den nächsten Satz auswählen kann. Je nachdem, wie Sie das Gespräch beginnen und auf die Antworten des Gesprächspartners eingehen, erhalten Sie nützliche Hinweise oder eine Abfuhr, weil Sie ihn durch irgendeine Frage beleidigt haben. Die Figuren auf dem Bildschirm reagieren richtig auf den Gesprächsverlauf und man kann ihnen deutlich ansehen, ob sie ärgerlich, verlegen oder interessiert sind.

In alle anderen Geschehnisse wird Sinbad durch Zufall verwickelt. So kann es passieren, daß das Schiff in gefährliche Gewässer kommt, wo Sie ihr seemännisches Geschick beweisen können, indem Sie Klippen umschiffen und Menschen rettten. An einigen Stellen lauern auch Piraten auf.

An Land warten ganz andere Gefahren. Bei einem Erdbeben muß man Felsspalten hochklettern und dabei herabfallenden Steinen ausweichen. An anderen Stellen greift der Schwarze Prinz an und fordert Sinbad zum Schwertduell. Auch ein Vogel macht Ihnen zu schaffen. Ihn erledigt man mit einem gezielten Schuß aus der Armbrust.

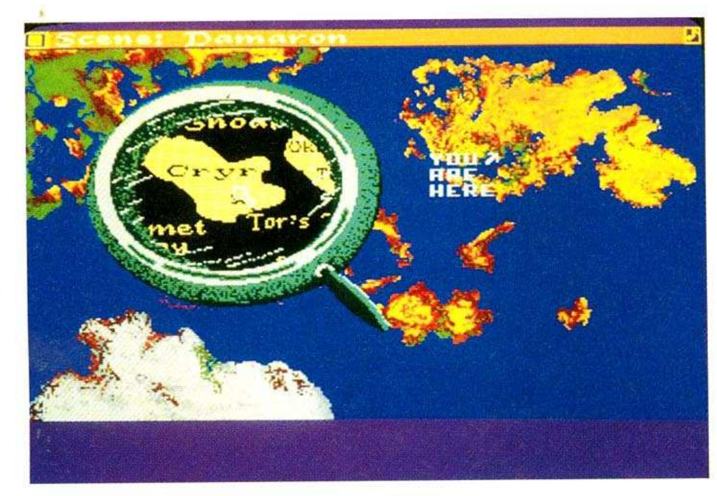
Auf einer Insel wartet eine besondere Überraschung. Ein Zyklop greift Sinbad an (Odysseus läßt grüßen!), der sich mit einer Steinschleuder zur Wehr setzt. Trifft man das Auge, ist der Riese blind und Sinbad hat gewonnen. Es gilt aber, schnell zu agieren, da der toll animierte Riese auch mit Steinen zurückwirft und nach einer Weile Sinbad trifft. Diese Sequenz gehört grafisch zu den besten des Programms.

Die Zwischenspiele auf der Reise sind aber nur eine Aufgabe für den Spieler. Es gibt drei Haupt-Bildschirme. Der erste ist eine ausführliche Landkarte der bekannten Welt. Sie gibt Auskunft über die Kontinente. Drückt man die linke Maustaste, erscheint eine fantastisch programmierte Lupe, die alle Einzelheiten auf der Karte, zum Beispiel die Lage von Orten, zeigt.

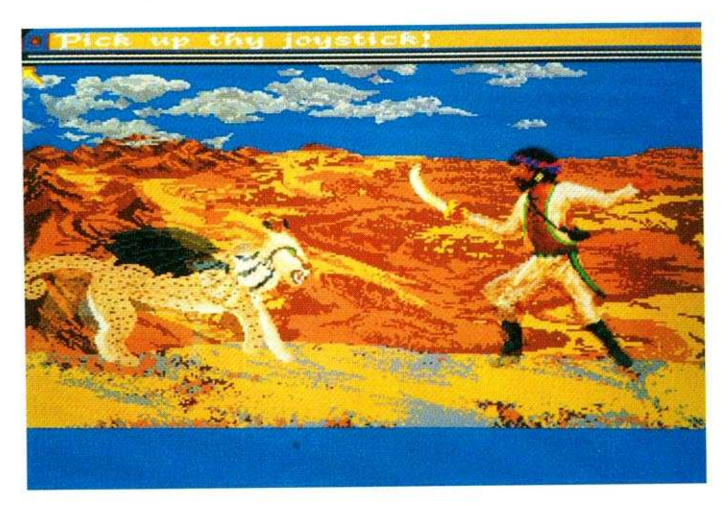
Der zweite wichtige Screen ist eine Sanduhr, bei der unablässig der Sand nach unten fällt und so zeigt, wie viel Zeit man noch hat, bevor der Kalif unwiderruflich für immer ein Falke bleiben muß.

Der dritte und letzte Bildschirm ist ein Taktikspiel. Der dunkle Prinz will bekanntlich das Reich des Kalifen erobern und steht mit seinen Truppen schon an den Grenzen. Während Sinbad durch die Welt zieht, muß man auf der anderen Karte die zurückgelassenen Truppen zum Schutz der Hauptstadt einsetzen. Entscheidende Bedeutung haben dabei die Versorgungszentren, in denen sich die Armeen besser ausrüsten und verstärken können. Wer diese Zentren kontrolliert, hat den Kampf so gut wie gewonnen.

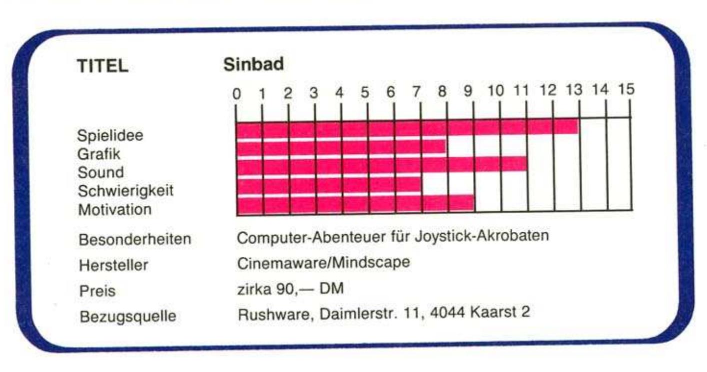
Sinbad ist ein Spiel, das sehr stark auf Effekte ausgelegt ist. Man hat sich alle Mühe gegeben, den Spieler mit grafischen Raffinessen und ungewöhnlichen Spielideen zu überraschen. Dabei hat man leider ein paar Kleinigkeiten nicht optimal hinbekommen. Zum Beispiel den Zoom-Effekt, der aus einer gut gezeichneten Figur eine wirre Ansammlung von großen farbigen Flecken macht. Trotz einiger grafischen Glanzlichter bleibt Sinbad hinter »Defender of the Crown« und »SDI« zurück. Es gibt zwar wesentlich mehr Spiel-Elemente, doch sie wirken insgesamt recht zusammengewürfelt und ergeben keine klare Linie. Was Abwechslung und Ideenreichtum angeht, hat Sinbad die Nase vorn. Spielerisch ist SDI jedoch eine Klasse besser, zumal Sinbad recht leicht zu lösen ist. (gn)



Die Weltkarte bei Sinbad unter der Lupe



Ein Schwertduell für wahre Joystick-Künstler



Rollenspiele mit Rasse und Klasse

Zwei der beliebtesten Fantasy-Rollenspiele liegen jetzt für den Amiga vor: »The Bard's Tale« und »Ultima III« im Vergleich.

Fantasy-Rollenspiele, das spannende Spielgenre, das beim C 64 zur Zeit einen Boom erlebt, sind auch beim Amiga ein Renner. Zwei Klassiker liegen jetzt auch in Amiga-Versionen vor: »The Bard's Tale« und »Exodus: Ultima III«. In unserem Vergleichstest achteten wir nicht nur auf die spielerischen Unterschiede, sondern analysierten auch, welches der beiden Spiele den Amiga besser ausnutzt.

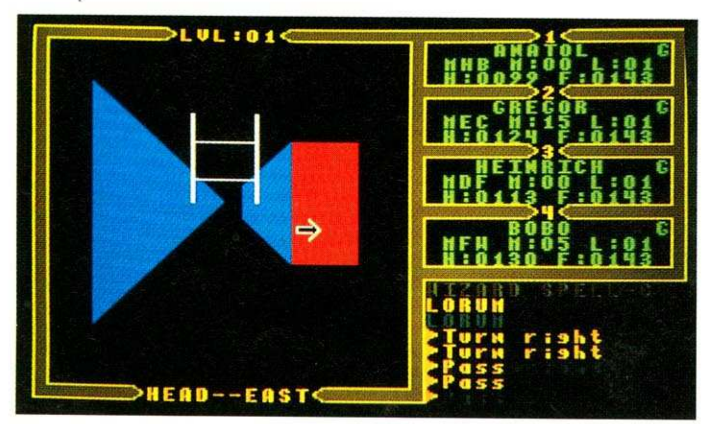
»The Bard's Tale« gilt heute schon als eines der erfolgreichsten Rollenspiele aller Zeiten, obwohl es erst Anfang 1986 für Apple II und C 64 erschien. Die Amiga-Umsetzung ist die dritte Version. Sie steuern eine Kampfgruppe (»Party«), die aus sechs Mitgliedern besteht. Am Anfang müssen Sie sich diese sechs Mannen noch zusammenstellen. Nach der Wahl der sieben Menschen- und zehn Berufsklassen richten sich die Charakter-Attribute Ihrer Kämpfer. Da Sie ja nur sechs Charaktere schaffen können, müssen Sie abwägen, für welche Berufe Sie sich entscheiden sollen.

Mit Ihren sechs Männern streifen Sie nun durch eine Stadt namens Skara Brae. Der Zauberer Mangar hat die ganze Ortschaft verhext. Es gibt keine braven Bürger mehr, sondern nur noch Kreaturen des Bösen: Werwölfe, Barbaren, Riesenspinnen, Golems und diverse andere nette Burschen. In Skara Brae gibt es eine ganze Reihe interessanter Gebäude, die teilweise auf dem Stadtplan verzeichnet sind, der dem Spiel beiliegt. In einer Schmiede decken Sie Ihre Kämpfer mit Waffen, Schildern und Rüstungen ein, in Kneipen gibt es Drinks und auch mal einen Geheimgang.

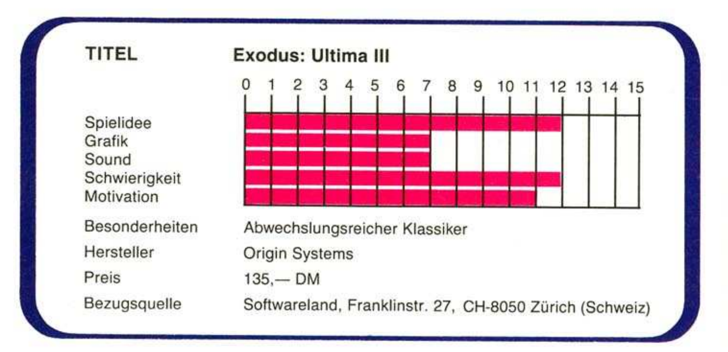
Während Sie durch die Straßen schlendern, werden Ihre Mannen oft von Monstern angegriffen. Das ist auch gut so, denn nur durch Siege im Kampf gewinnen die Charaktere Erfahrungspunkte (»Experience Points«). Wenn ein Charakter eine bestimmte Anzahl Experience Points gesammelt hat, kann er vor einem Tribunal namens »Review Board« vorsprechen. Hier steigt der Charakter dann einen Level auf. Das bedeutet, daß er mehr Hitpoints erhält und damit mehr Schläge im Kampf einstecken kann, ohne zu sterben. Außerdem kann sich eine der fünf Charakter-Eigenschaften um einen Punkt verbessern. Wenn Ihre Charaktere stark geworden sind und die Magier neue Zaubersprüche dazugelernt haben, geht das Spiel aber erst richtig los. Unter Skara Brae gibt es sage und schreibe 15 Labyrinthe (Dungeons) und jedes davon ist so groß wie die ganze Stadt! Hier trifft man die wirklich fiesen Monster und andere Schikanen. Rätsel, Teleporter- und Dunkelzonen, Räume, in denen die Magie nicht wirkt und Fallen gehören zum Dungeon-Alltag.



Kampf gegen acht Monster (Ultima III)



Eine Leiter führt aus dem Dungeon heraus (Ultima III)



TITEL	Exodus: Ultima III
Spielidee	12
Grafik	7
Sound	7
Schwierigkeit	12
Motivation	11
Besonderheiten	Abwechslungsreicher Klassiker
Hersteller	Origin Systems
Preis	135,— DM
Bezugsquelle	Softwareland, Franklinstr. 27, CH-8050 Zürich (Schweiz)

Aber man muß sich durchkämpfen, um am Schluß schließlich in einem großen Finale Mangar gegenüberzustehen.

Die »Ultima«-Serie, die von einem amerikanischen Programmierer mit dem schönen Künstlernamen Lord British stammt, gehört zu den unbestrittenen Klassikern des Rollenspiel-Genres. »Exodus: Ultima III« nennt sich das bereits dritte Ultima-Spiel. Sie steuern hier eine Party, die bis zu vier Charaktere haben kann. Die Grundeigenschaften der Charaktere (Stärke, Ausdauer, Intelligenz und Weisheit) bestimmen Sie selber. Im Gegensatz zu The Bard's Tale spielt der Zufall bei der Erschaffung eines Charakters keine Rolle. Sie dürfen bei Ultima III eine bestimmte Anzahl von Punkten zwischen den vier Eigenschaften verteilen. Die Wahl von Charakter-Klasse und Rasse ist auch wichtig, denn davon hängt ab, welche maximale Punktzahlen eine Spielfigur bei den Charakter-Eigenschaften im Lauf des Spiels erringen kann. Natürlich gibt es noch weitere Feinheiten: Einige Waffen und Rüstungen können wie bei The Bard's Tale nur von bestimmten Klassen benutzt werden, bei Druiden regenerieren sich die Magie-Punkte am schnellsten und so weiter.

Eine Hintergrundgeschichte gibt's natürlich auch, die an das Vorgängerspiel »Ultima II« anknüpft. In diesem Spiel wurde ein Oberschurke namens Minax beseitigt und das Fantasy-Land (in dem Lord British sich selbst als guter König verewigt hat) lebt in Frieden und Freundschaft. Nach 20 Jahren hat diese schöne Zeit aber ein jähes Ende. Wie sich langsam herausstellt, hatte Minax einen Sprößling namens Exodus, der mittlerweile zum Oberschurken herangewachsen ist.

Lord British, der besorgte König, befürchtet das Ende der Welt. Er bittet um die Hilfe mutiger Helden, die das Unmögliche schaffen sollen: Exodus finden und vernichten.

Ihre Reise ins Ungewisse führt Sie über das Land, auf hohe See, in Städte, Schlösser und unterirdische Labyrinthe. Kämpfe mit Monstern stehen auch hier an der Tagesordnung, sind aber nicht so sehr im Mittelpunkt des Geschehens wie bei The Bard's Tale. Mit Zaubersprüchen und Waffen geht es den Gegnern an den Kragen; sogar die Entfernung zu den Angreifern spielt

eine Rolle. Das Repertoire an Zaubersprüchen ist allerdings nicht so umfangreich wie beim Konkurrenzspiel.

Ein wesentlicher Punkt beim Vergleich dieser beiden Programme ist die Frage, wie gut der Amiga ausgenutzt wird. In dieser Hinsicht hat The Bard's Tale klar die Nase vorn, obwohl Ultima III eine Reihe von Verbesserungen gegenüber der C 64-Version bietet. Viele Funktionen können mit der Maus gesteuert werden, manchmal kommt man aber nicht um eine Tastatureingabe herum. Die Musik und die spärlichen Soundeffekte klingen eine Klasse besser. Die Grafik ist zwar schneller und etwas bunter geworden, wird einem Prachtcomputer wie dem Amiga aber nicht ganz gerecht.

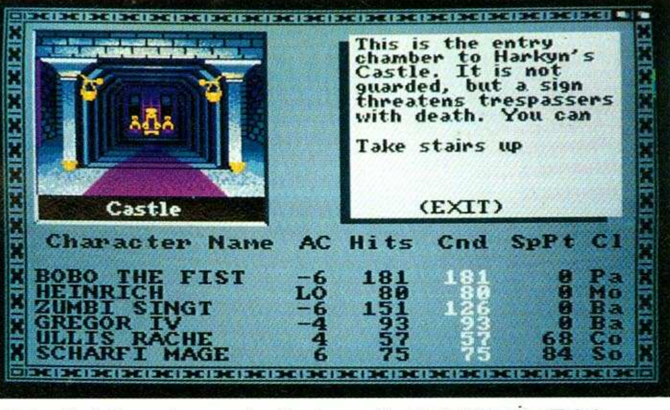
Ganz anders The Bard's Tale, das technisch zu den besten Amiga-Spielen zählt, die es zur Zeit gibt. Sound und Musik kommen zwar nur vereinzelt vor, sind aber dann von besonderer Güte. Wenn der Barde ein Lied spielt, bekommt man das natürlich zu hören und auch in der Gilde, in der jedes Spiel beginnt, ertönt eine gut komponierte Melodie. Einen Sonderpreis aber für den digitalisierten Sound, mit dem man beim Besuch eines Tempels verwöhnt wird: Nachdem ein Charakter geheilt wurde, gibt ein sehr eindrucksvoller Chor ein kleines Liedchen zum Besten — wohlgemerkt nicht gespielt, sondern richtig gesungen!

Die Grafik ist noch zwei Klassen besser. Hier hat man nicht einfach die Bilder der C 64-Version übertragen und etwas aufgepeppt; vielmehr wurden alle Grafiken völlig neu gemalt. Die Animation, die man schon vom C 64 her kennt, wurde wesentlich verbessert. Ein Zombie ist bei der Amiga-Version von The Bard's Tale kein besseres Strichmännchen à la Ultima, sondern ein hervorragend gemalter, ausgesprochen gruseliger Geselle. Die Bildschirmfotos auf diesen Seiten sagen aber am meisten über die Qualität der Grafik aus.

Ein dickes Lob auch für die Benutzerführung. Garantiert jeden Befehl kann man über die Maus eingeben, was einen besseren Spielkomfort als bei Ultima III bewirkt. Man muß sich aber nicht von den Tasten trennen: Wer will, kann wie beim C 64 alle Kommandos auch per Tastatur eingeben.

Welches Programm vom spielerischen Standpunkt her besser ist, kann man nicht so ohne weiteres sagen. Aus Leserzuschriften wissen wir, daß es Ultima-Fans gibt, die für The Bard's Tale nicht viel übrig haben. Es gibt aber auch eine sehr große Gruppe Bard's-Tale-Fans, die wiederum ihren Liebling allen anderen Fantasy-Rollenspielen vorziehen. Ich persönlich finde das Spielprinzip von The Bard's Tale faszinierender und motivierender als bei Ultima III. Ultima-Fans mögen mich wegen dieser Aussage nicht mit einem bösen Zauber belegen, aber The Bard's Tale ist besser aufgebaut, nicht so verwirrend, gradliniger zu spielen und ganz einfach eine Spur reizvoller. Das liegt auch zu einem wesentlichen Teil an der Überlegenheit, was Grafik, Sound und Benutzerführung angeht. Wer Ultima III auf dem C 64 kennt und sich Wunderdinge von der Amiga-Version verspricht, wird etwas enttäuscht sein. Die Amiga-Umsetzung von The Bard's Tale hingegen ist eine ganz wesentliche Steigerung gegenüber dem Original und nutzt den Computer gut aus. Selbst wenn Sie dieses Spiel schon auf dem C 64 kennen, werden Sie von der Amiga-Version aufs neue fasziniert sein.

Der Sieger nach Punkten in unserem Fantasy-Vergleichstest heißt also The Bard's Tale. Dazu kommt noch der Preisfaktor, denn Ultima III ist ein Stück teurer, bietet dafür aber eine bessere Aufmachung (drei Handbücher und eine Stoffkarte, die man mit viel gutem Willen auch als Tischdecke benutzen kann). Das soll aber nicht heißen, daß Ultima III schlecht ist. Es ist vielmehr ein gutes Programm, aber The Bard's Tale ist eben noch eine Klasse besser. (hl)



Viele Gefahren lauern in Harkyns Schloß (Bard's Tale)



Ein Ogre greift an, doch »Bobo the Fist« wehrt sich



Der Amiga im Fernsehstudio

Sowohl im privaten Heimstudio als auch im Fernsehstudio wird der Amiga heute als Grafikcomputer gezielt eingesetzt. Anhand der Fernsehsendung »Computerzeit« werden wir Ihnen den Einsatz eines Computers in der Praxis zeigen.

Mit der Computerzeit fing alles an. In der Anfangszeit dieser Sendereihe stand man ständig vor dem Problem, ein qualitativ hochwertiges Computerbild in die Fernsehstudio-Anlage einzuspeisen. Es wurden viele Computersysteme der verschiedensten Hersteller getestet. Aber immer wieder war das Ergebnis unbefriedigend, bis man eines Tages auf den Amiga 1000 von Commodore stieß. Er erwies sich sehr schnell als das »Ei des Columbus«. Denn neben seinen hervorragenden grafischen Eigenschaften stellt er ein FBAS-PAL-Signal (Farbfernsehsignal nach der PAL-Norm) zur Verfügung, das ohne technischen Aufwand an das Studiosystem angeschlossen werden konnte. Genau so, wie wir den Amiga an der Studioanlage betreiben, kann der Hobby-Video-Filmer seinen Amiga an einen Videorecorder anschließen und Grafiken des Computers aufzeichnen.

Viele werden sich nun fragen: Wozu ein Computer in einer Fernsehsendung? Kann man die Informationen nicht anders vermitteln?

Das Fernsehen ist ein audiovisuelles Medium, das heißt, daß Sprache und Bild eine unzertrennliche Einheit bilden. Aus diesem Grund wird das gesprochene Wort durch ein geeignetes Bild unterstützt. In den meisten Fällen kann ein reales Bild mit einer Nachricht verknüpft werden (Bild 1). Das können Filmzuspielungen oder auch Demonstrationen im Studio sein. Ein Beispiel soll das verdeutlichen:

In den täglichen Nachrichten wird berichtet, daß ein großer deutscher Automobilhersteller ein neues Fahrzeug der Öffentlichkeit vorgestellt hat. Aber wie sieht es aus, und was leistet es? Der Nachrichtensprecher kann nun mehrere Minuten über Farbe, Form und technische Daten sprechen, aber der Zuhörer kann sich das Fahrzeug trotz ausführlicher Beschreibung immer noch nicht bildlich vorstellen. Anders ist es, wenn man die Einzelheiten visuell, also in einem Film oder auf einem Bild zeigt.



Bild 1. Fernsehstudio des Bayerischen Rundfunks in München-Unterföhring. Das Fernsehteam bei der Aufzeichnung einer Musiksendung aus der Serie »Computerzeit«.

Solange es sich um reale Gegenstände handelt, ist es recht einfach, passende Bilder oder Filme zum Text zu finden. Wie ist es aber mit irrealen Objekten?

Solche Problemstellungen können nur durch Symbole, Zeichnungen oder Skizzen verdeutlicht werden. Und damit sind wir wieder bei unserem Grafikcomputer angekommen, denn genau hier findet er seinen Einsatz in der Programmgestaltung. Mit dem Amiga lassen sich einfach und schnell erklärende Grafiken zeichnen.

Sehen wir uns dazu ein Beispiel aus der Praxis an.

Problemstellung:

In der Folge 21 der Computerzeit (mit dem Thema Robotik) wird die Leistungsfähigkeit der heutigen Industrieroboter gezeigt. Die Besonderheit die-

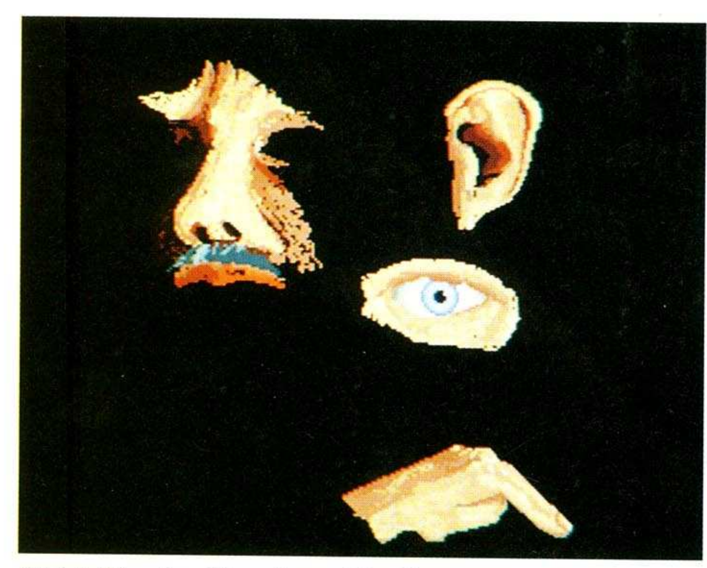


Bild 2. Einzelgrafiken Auge, Ohr, Nase und Hand mit fühlendem Finger; wichtige Sinnesorgane für Roboter

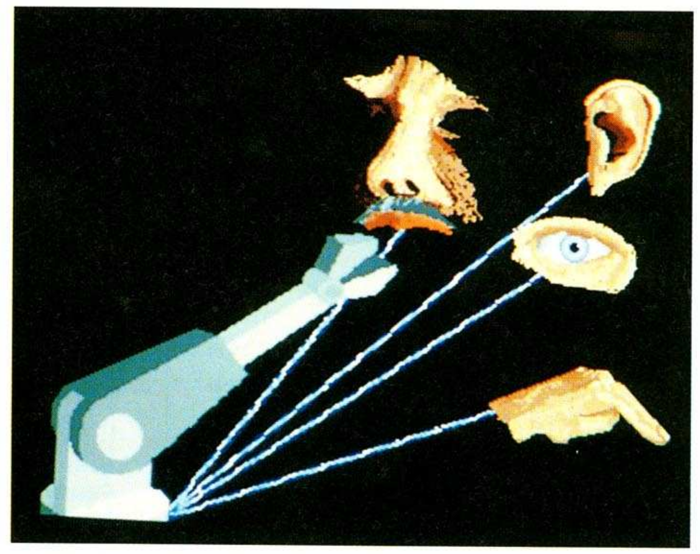


Bild 3. Das fertige Bild mit dem Roboterarm, den Sensoren und den wichtigen »Datenleitungen« für die Verbindungen

ser neuen Roboter sind ihre weiterentwickelten Sensoren. Wie diese Fühler einen Roboterarm steuern können, soll durch eine Grafik verdeutlicht werden. In der folgenden Beschreibung werden kurz die Vorarbeiten zur Lösung eines solchen Problems in einzelnen Schritten erklärt.

Erster Schritt:
Mit einem Zeichenprogramm, in diesem Fall DeluxePaint, werden die Sensoren als Sinnbilder dargestellt.

Es wird mit der Symbolisierung der Bilderkennung (das Sehen) begonnen. Hierzu wird das Abbild eines menschlichen Auges gewählt. Die naturgetreue Wiedergabe des Auges wurde hierzu bewußt eingesetzt, um die hohe Leistungsfähigkeit eines Bilderkennungssystems in den Vordergrund zu stellen.

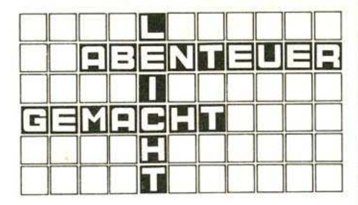
Ebenso wie das Symbol Auge für den Sensor »Sehen« (Graustufenkamera) verwendet wurde, wird das Fühlen einer Hand mit einem tastenden Finger, das Riechen einer Nase und das Hören einem Ohr gleichgesetzt (Bild 2). So erkennt man einfach und schnell, daß Sensoren die Sinnesorgane eines Roboters sind.

Zweiter Schritt:
Das nächste Bild der Grafik soll einen Industrieroboter zeigen. Er steht für das ausübende Organ in dem System. Um die Unvollkommenheit der technischen Nachbildung eines Arms hervorzuheben, wird eine möglichst einfache Darstellung gewählt.

Dritter Schritt:
Der letzte Schritt im Ablauf ist, den Datentransfer zwischen den Sensoren und dem Roboter zu versinnbildlichen. Hier wird eine besondere Eigenschaft des Amigas genutzt. Sie liegt darin, Farben in einen definierten Bereich im Rhythmus zu wechseln. Dieser Farbdurchlauf erzeugt die Illusion einer Bewegung, eben die der laufenden Dateninformation zwischen Sensor und Roboterarm (Bild 3).

Bei den Arbeiten zur Fernsehsendung werden besonders die Programme Deluxe-Paint und Deluxe-Video eingesetzt, weil mit ihnen eine sehr hohe Flexibilität und ein einfaches Zeichnen von Grafiken möglich ist. Außerdem lassen sich mit Deluxe-Video Animationen entwerfen, um Bewegungsabläufe zu zeigen.

(do)



BORROWED TIME
Bevor Farnham überführt werden kann, muß Fred Mongo der Polizei übergeben werden.

Die Gangster auf den Fersen. Rein ins Hotel und hinter den Sessel.

Eine Brieftasche ist wichtig im Leben, doch nicht für Detektive.

Wer zweimal den Revolver zeigt, macht doppelt soviel Eindruck.

Es lohnt sich, Gespräche zu belauschen.

Milli, Doris und Bruce haben etwas gemeinsam.

TASS TIMES IN TONETOWN
Franklin Snarl wird man nur in Verkleidung los. Mit Anzug, Maske, Halsreif und Federn im Haar sieht man richtig aus.

Zur Frau vorm Konzert darf man unfreundlich sein, nur so erfährt man ihren Vornamen.

Im Pet-Shop lohnt sich das Warten auf Franklin. Es gibt etwas Wichtiges zu kaufen.

Zeitung lesen informiert.

Das Licht im Glas wird an manch dunkler Stelle helfen.

THE PAWN
Whisky kaufen ist angebracht, selber trinken jedoch nicht.

Die Felsbrocken müssen weg. Zwei längere Werkzeuge und ein Kleidungsstück. Das ist wohl der ganze Trick.

Das Ziel des Abenteuers ist doch klar: Leg' alle Bande ab und kehre in die eigene Welt zurück.

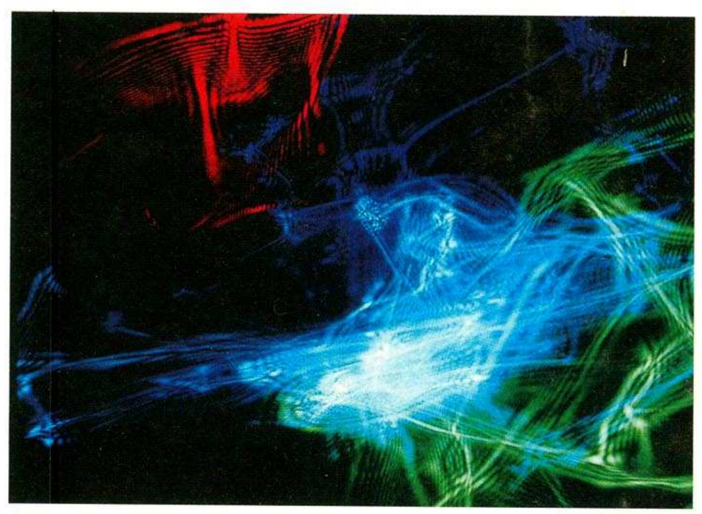
Kronos hat weit mehr bei sich als nur den Zettel. Es lohnt sich, übers Armband zu fragen.

Der Baumstumpf verbirgt etwas, doch erst der Guru bringt es zutage.

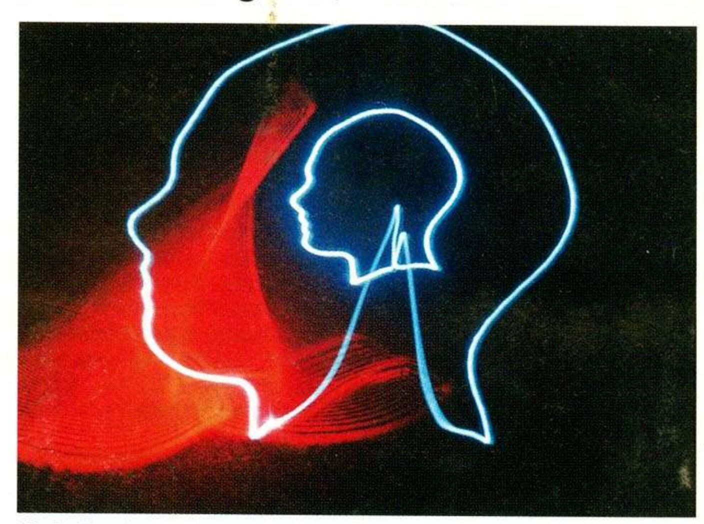
Vorsicht vor dem »Adventurer«, er kann tödlich sein. Doch ein Geschenk ist auch für ihn tödlich.

Laser in Perfektion

Bei Laserland werden Lasershows programmiert und zu verschiedenen Anlässen, zum Beispiel bei Messen wie der Electronica, vorgeführt. Computer wurden dafür schon immer eingesetzt, aber mit dem Amiga eröffnen sich vollkommen neue Dimensionen für die Lasersteuerung in bezug auf Geschwindigkeit und Farbe.



Schöne Effekte mit vielen Farben



Viele Punkte und zwei Farben

Zu einer guten Lasershow gehört außer Kreativität auch das passende Werkzeug. Mit Werkzeug ist hier der Computer gemeint, er dient zur Programmierung der gesamten Show und zur späteren Vorführung. Der Amiga ist aufgrund seiner Geschwindigkeit und seiner Grafik wie geschaffen für diese Anwendung. Was kleinere Computer nicht mehr schaffen, ist mit dem Amiga realisierbar, flackerfreie Bilder mit mehr als 1000 Punkten.

Der erste Schritt auf dem Weg zur guten Lasershow ist das Entwerfen der einzelnen Bilder. Damit diese Bilder später auch gut wirken, müssen sie mindestens eine Auflösung von 256 x 256 Punkten haben. Bei dem Programm Laser-Pac wird, auf dem Bildschirm, mit der mittleren Auflösung von 640 x 256 Punkten gearbeitet, da sich rechts neben der Fläche für das Bild noch einige Gadgets befinden. Gezeichnet wird natürlich mit der Maus, wobei man immer von dem zuletzt gesetzten Punkt eine Linie zur Mausposition zieht. Die Punkte werden gespeichert, indem die Koordinaten und die passende Farbe in einer Liste abgelegt werden. Der Grund für diese Form der Speicherung liegt in der Natur des Lasers. Die Ansteuerung der zwei Ablenkspiegel muß nämlich mit einem X- und Y-Wert für die beiden Achsen erfolgen. Verschiedene Farben lassen sich hier auch einstellen, sie werden dann später bei der Vorführung auch benutzt.

Werden nun mehrere Bilder gezeichnet, indem man sie schrittweise verändert, können ganze Bewegungsabläufe erzeugt werden. Aber für komplexe Bewegungen genügt das nicht, deshalb gibt es einen Befehl, mit dem man festlegen kann, auf welcher Bahn sich die Bilder bewegen sollen. Die Eingabe dieser Bahn erfolgt praktisch wie bei einem Bild durch das Setzen einzelner Punkte. Es ist sogar möglich, ein fertiges Bild so einzusetzen, daß sich ein anderes an dessen Kanten entlangbewegt.

Viele Farben

Diese Funktionen könnte man sicherlich auf vielen anderen Computern auch verwirklichen, aber die folgenden Funktionen laufen nur auf einem wirklich schnellen Computer wie dem Amiga. Ein Bild läßt sich durch Zoomen beliebig vergrößern und verkleinern. Man sieht auf dem Bildschirm wie aus einem Punkt die Figur auftaucht und immer größer wird und sich dann um sich selbst dreht. Das Fantastische daran ist jedoch, daß der angeschlossene Laser die Bewegung auf der Leinwand nachvollzieht. Die Geschwindigkeit ist dabei natürlich nicht ganz so hoch, wie später bei der Vorführung, da jeder Schritt zu diesem Zeitpunkt erst berechnet werden muß. Trotzdem ist es faszinierend mit anzusehen, wie Figuren auf den vorgeschriebenen Bahnen über die Leinwand wandern.

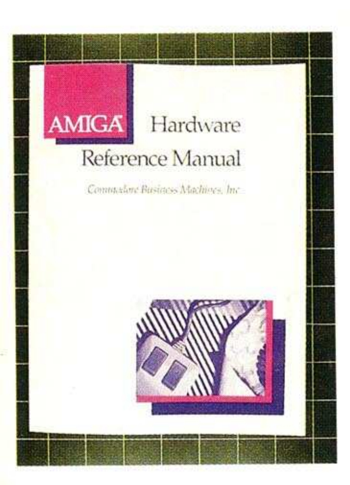
Fantastische Effekte erreicht man auch, indem man der eigentlichen Figur noch eine Schwingung unterlegt. Zum Beispiel sehen Bilder, die mit einer Sinusschwingung überlagert werden, aus, als ob sie auf dem Wasser schwimmen. Diese Schwingungen werden hier mit einem kleinen Trick erzeugt: Man läßt den Soundchip mit einstellbarer Frequenz und Amplitude eine Schwingung vollziehen und übernimmt dann die Werte.

Sollen die Bewegungen in einer Lasershow ruckfrei erscheinen, müssen die einzelnen Bilder mindestens 25mal in der Sekunde projiziert werden. Wenn die Bilder auch komplexe Formen haben dürfen, sind mehr als 1000 Punkte pro Bild erforderlich. Bei der Vorführung können die Bilder auf der Leinwand mit einer Auflösung von 4096 x 4096 Punkten bewegt werden. Das heißt, daß im Computer mit 12 Bit gerechnet werden muß. Für jeden Punkt müssen zwei Koordinaten und die Farbe geschickt werden. Rechnet man nun aus, wieviel Zeit für die Ansteuerung eines einzelnen Punktes zur Verfügung steht, erschrickt man zunächst. Das Übertragen eines 12-Bit-Wertes muß in nur 1/75 000 Sekunde geschehen. Aber durch den geschickten Aufbau des D/A-Wandlers ist es möglich. Der Wandler wird an den Systembus des Amiga angeschlossen und mit Schaltern auf eine bestimmte Adresse eingestellt. Will man nun einen Wert an den Laser ausgeben, beschreibt man einfach die voreingestellte Adresse mit dem passenden Wert.

Ganz neue Entwicklungen auf dem Gebiet der Lasertechnik will Laserland in nächster Zukunft einsetzen. Bei dem neuen System wird ein Gerät an den Videoausgang des Computers angeschlossen. Die Signale werden dann in Impulse für mehrere Laser umgesetzt, die den Inhalt des Bildschirms auf eine bis zu 15 Meter breite Leinwand projizieren. Auf diesem Hintergrund laufen dann die vom Amiga berechneten Lasershows ab. Ein zusätzlicher Reiz ist die Musik, die zur Show abgestimmt, dazu gespielt wird. Wenn Sie also von einer solchen Vorstellung hören, überlegen Sie nicht lange, sondern machen Sie sich sofort auf den Weg. Wer eine eigene Show braucht, kann sie sich von Laserland entwickeln und dann vorführen lassen.

(rb)

Laserland, Wölkham 1, 8201 Halfing, Telefon 08 55/15 51



Hardware Reference Manual

Dieses Buch wendet sich an denjenigen, der den Amiga an sich schon beherrscht, aber mit direktem Zugriff auf die Custom-Chips noch mehr erreichen will. Nach einer kurzen, allgemeinen Einführung in die Hardware des Amiga, wird als erstes der Copper genauestens unter die Lupe genommen. Die drei Anweisungen, die der Copper versteht, mit ihren Anwendungen werden anhand von Beispielen und Anwendungen erkärt. Auch das Zusammenspiel mit dem Prozessor und dem Blitter sind hier erwähnt.

In den nächsten zwei Kapiteln steht alles Wissenswerte über die Anwendung von Playfields und Sprites. Egal ob einfaches oder doppeltes Playfield, alle Möglichkeiten werden exakt beschrieben. Bei den Sprites wird vom Datenformat über das Verketten zweier Sprites bis zum mehrfachen Verwenden eines Sprites alles angesprochen, was von Interesse ist. Wie im gesamten Buch sind auch hier die Beispielprogramme in Assembler geschrieben.

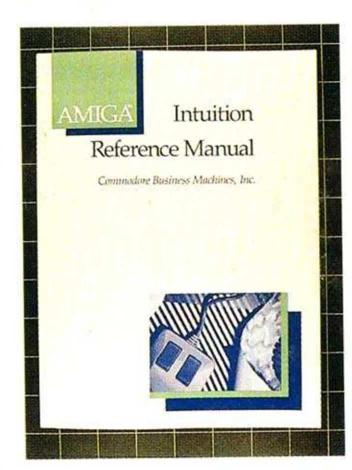
Doch auch die Musikfans kommen nicht zu kurz, denn der Tonerzeugung ist ein eigenes Kapitel gewidmet. Sogar der Einsatz von DMA-Kanälen wird beschrieben.

Der Blitter, einer der interessantesten Teile im Amiga, ist zuständig für die Verschiebung großer Datenblocks. Seine Fähigkeiten und das nötige Wissen zu seiner Anwendung werden so gut erklärt, daß man wirklich Nutzen daraus ziehen kann.

Es folgt ein Kapitel über die Prioritäten von Playfields, Sprites und der Interrupts, das für manche Anwendung sehr wichtig sein kann.

Im letzten Kapitel werden dann die Schnittstellen bis ins Detail beschrieben. Die Angaben beziehen sich allerdings nur auf den Amiga 1000 und sind somit teilweise für Benutzer des Amiga 500 oder 2000 nicht relevant. In den Anhängen erfolgt eine Zusammenfassung aller Daten, in der man gut nachschlagen kann, wenn man keine genaue Beschreibung mehr benötigt. (rb)

Hardware Reference Manual, Addison-Wesley GmbH, 264 Seiten, ISBN 0-201-11099-7, 24,95 US-Dollar



Intuition Reference Manual

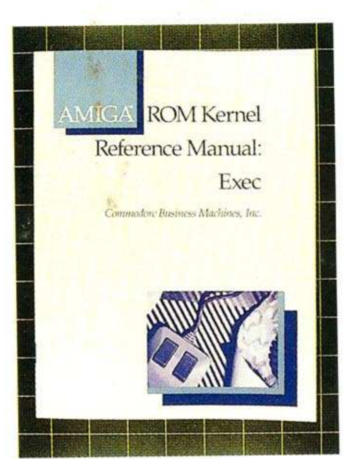
Dies ist das erste Buch der Reihe, das man lesen sollte, wenn man den Amiga ausnutzen will. Als erstes findet man eine allgemeine Einleitung über die Funktion von Intuition. Doch dann wird es sofort interessant, es wird ein kleines Programm gezeigt, das zunächst nur mit einem Fenster arbeitet. Nach und nach wird das Programm dann ausgebaut, so daß am Schluß alle grundlegenden Funktionen von Screens und Windows enthalten sind.

Das bisher Gelernte wird nun in den einzelnen Kapiteln ausführlicher behandelt. Sehr wichtig für den Programmierer ist in diesem Zusammenhang die genaue Beschreibung der einzelnen Elemente der Struktur von Screens. Nun folgt im Buch der nächste logische Schritt: die Beschreibung von Windows. Mit den Informationen, die hier geboten werden, ist es nicht mehr schwer, eigene Fenster mit beliebigen Attributen zu versehen.

Nachdem die Systemgadgets beschrieben wurden, kommt der interessante Teil dieses Kapitels, die benutzerdefinierten Gadgets. Nach der Beschreibung des Aufbaus und der Struktur von Menüs folgt dann der Teil über Requester. Der nächste Teil ist hauptsächlich für die Grafik-Freaks von Bedeutung, es geht um Grafikfunktionen, Text und Bilder.

Ein Buch, das jeder Amiga-Freak zu Hause haben sollte, wenn er seinen Amiga selber erfolgreich programmieren will.

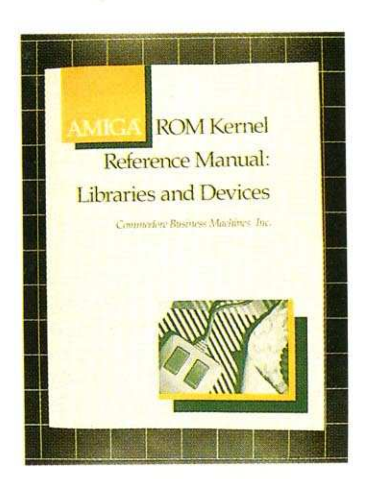
Intuition Reference Manual, Addison-Wesley GmbH, 358 Seiten, ISBN 0-201-11076-8, 24,95 US-Dollar



ROM Kernel Reference Manual: Exec

Für alle, die sich mit den Multitasking-Fähigkeiten des Amiga beschäftigen wollen, ist dieses Buch die wichtigste Informationsquelle. Die Struktur von Listen und die zugehörigen Funktionen zum Ändern von Listen können hier erlernt werden. Für das Verständnis des folgenden Kapitels, das von Tasks handelt, wird vorausgesetzt, daß man schon einen gewissen Einblick in Multitasking-Systeme hat. Die Erklärung, wie man einen Task erzeugt, wird durch ein Beispielprogramm ergänzt, was viel zum Verständnis beiträgt. Es folgt ein sehr wichtiges Thema, der Austausch von Messages (Nachrichten) zwischen den Tasks. Auch wer Interrupts verwenden will, findet im Buch die nötigen Informationen über dieses komplexe Thema. Wer zu den vorhandenen Bibliotheken noch eigene hinzufügen will, kann sich die benötigten Informationen im folgenden Kapitel holen. Den Abschluß bildet die Beschreibung des im ROM vorhandenen Programms ROM-Wack. (rb)

ROM Kernel Reference Manual: Exec, Addison-Wesley GmbH, 260 Seiten, ISBN 0-201-11099-7, 24,95 US-Dollar



ROM Kernel Reference Manual: Libraries and Devices

In der ersten Hälfte des Buches werden die Grundlagen für Grafik erklärt. Zuerst behandelt das Buch die Playfields und deren Einsatz. Natürlich kommt die genaue Beschreibung der Struktur und der Funktionen nicht zu kurz. Anhand eines Grafik-Beispielprogramms, das wie alle Listings gut dokumentiert ist, kann das Erlernte dann nachvollzogen werden. Im folgenden Kapitel geht es um die Layers, das sind Grafikbereiche, die sich gegenseitig überlagern können. Alle benutzbaren Funktionen sind aufgezählt und erläutert. Das Öffnen der Bibliothek für Layer und deren Benutzung schließen dieses Kapitel ab. Nun folgt ein Teil, in dem sich alles um die Animation von Sprites und Bobs dreht. Die Benutzung von Text, verschieden Zeichensätzen und Schriftarten wird anschließend behandelt.

Die nächsten 12 Kapitel beschäftigen sich mit den verschiedenen Devices. Hier findet man alles, was man zu deren Bedienung benötigt. Vom seriellen über den parallelen Port bis zur Behandlung der Laufwerke sind sie alle aufgeführt. Ein für viele interessantes Thema folgt: die Mathematik-Bibliotheken. Die beiden Formate, IEEE und FFP, sind gut erklärt und können nach dem Lesen leicht eingesetzt werden. Vor den Anhängen steht dann noch ein Abschnitt über die Workbench. In den Anhängen kann man die wichtigen Fakten schnell finden, wobei die Erklärungen nicht mehr so ausführlich sind. (rb)

ROM Kernel Reference Manual: Libraries and Devices, Addison-Wesley GmbH, 1024 Seiten, ISBN 0-201-11078-4, 34,95 US-Dollar

Wir suchen das beste Bild auf dem Amiga

Der Amiga ist ein Computer, dessen Stärken unter anderem auch in der Grafik liegen. Wir suchen das schönste »Gemälde«. Machen Sie mit — 1000 Mark winken!

In Sachen Grafik ist der Amiga derzeit wohl ungeschlagen: Mit seinen insgesamt 4096 Farben, die er darstellen kann, den verschiedenen Auflösungen (320 x 200 bis 640 x 400 Punkte) und den außergewöhnlichen Zeichenprogrammen, die dies unterstützen, kann ihm kein anderer Computer dieser Größenordnung so schnell das Wasser reichen. Deshalb starten wir in dieser Ausgabe einen neuen Wettbewerb, bei dem sich jeder angesprochen fühlen soll, der einen Amiga und ein Malprogramm besitzt.

Gesucht ist das schönste farbige Bild. Uns ist es egal, ob es in 320 x 200, 320 x 400, 640 x 200 oder 640 x 400 Punkten Auflösung erstellt wurde. Die beiden Bedingungen, die wir stellen, sind:
1. daß das betreffende Bild im IFF-Format, das beispielsweise von Deluxe-Paint I/II, Aegis Images oder Graphicraft erzeugt wird, vorliegt und
2. wäre es wünschenswert, wenn das Bild gezeichnet und nicht ganz und gar digitalisiert ist, denn das wäre zu leicht.



Teilbereiche des Bildes dürfen aber durchaus auch von einer Kamera oder einem Videorecorder übernommen werden (wenn man etwa eine besonders schöne Aufnahme einbinden möchte).

Als Preis winken insgesamt 1000 Mark, die wie folgt auf die ersten drei Gewinner verteilt werden:
1. Preis: 600 Mark
2. Preis: 250 Mark
3. Preis: 150 Mark

Bei genügend schönen Bildern könnte auch in Erwägung gezogen werden, eine Bilder-Diskette mit den besten Gemälden zu produzieren. Schreiben Sie uns doch, unabhängig von diesem Wettbewerb, ob Sie Interesse an einer solchen Diskette haben. Würden Sie sich eine Bilder-Diskette kaufen, und was würden Sie dafür ausgeben wollen? Sie helfen uns damit, auf Ihre Wünsche zu reagieren und Ihnen Produkte anzubieten, die auf Ihre Ansprüche zugeschnitten sind.

Schicken Sie Ihr Bild auf Diskette an folgende Anschrift (vergessen Sie bitte nicht, eine unterschriebene Copyright-Erklärung für das Werk beizulegen):

Markt & Technik Verlag AG
Redaktion Amiga-Magazin
Wettbewerb: IFF-Bild
z. Hd. Herrn Dieter Mayer
Hans-Pinsel-Str. 2
8013 Haar bei München

Einsendeschluß ist der 14. Juli 1987 (Datum des Poststempels). Jeder kann mitmachen und gewinnen. Ausgeschlossen sind Mitarbeiter des Verlages und deren Angehörige.

(dm)

Programmierer aufgepaßt: 1000 Mark Belohnung!

Haben Sie schon einmal ein Listing aus einer Computerzeitschrift abgetippt? Welch frustrierender Moment, wenn man nach mehrstündiger Arbeit feststellt, daß das Programm nicht funktioniert. In den allermeisten Fällen hat man doch noch irgendwo einen Tippfehler gemacht...

Aber das muß nicht sein. Ein Beispiel: Für den Commodore 64 setzt unsere Schwesterzeitschrift 64'er-Magazin schon seit längerer Zeit erfolgreich eine Eingabehilfe, den »Checksummer« ein. Dieser funktioniert so: Im Listing steht neben jeder Programmzeile eine Prüfsumme. Als Leser aktiviert man vor dem Abtippen eines Listings das Checksummer-Programm, das dann unsichtbar im Hintergrund arbeitet. Wenn man nun eine Programmzeile eintippt und die RETURN-Taste drückt, wird sofort auf dem Bildschirm die Prüfsumme für diese Zeile ausgegeben. Ein Blick in die Zeitschrift sagt einem dann sofort, ob man die Zeile fehlerfrei abgetippt hat.

Auch für den Amiga suchen wir nun einen solchen Checksummer. Eine echte Herausforderung für jeden Programmierer! Darum unser Aufruf: Schreiben Sie einen Checksummer für den Amiga. Für das beste Programm-System zahlen wir Ihnen 1000 Mark!

Ein paar Ideen zum »Warmdenken«:
— Man könnte einen Task im Hintergrund ablaufen lassen, der beim Drücken von <RETURN> automatisch eine Prüfsumme über die zuletzt eingegebene Zeile errechnet und sofort anzeigt.
— Etwas aufwendiger, aber nicht minder genial: Ein neuer, komfortabler Editor mit eingebautem Checksummer.
— Eventuell könnte man den Checksummer auch so gestalten, daß man für jede Zeile die Prüfsumme aus dem Heft mit abtippen muß. Dafür ließe sich aber (ein entsprechender Prüfalgorithmus vorausgesetzt) anhand eines programmgesteuerten Vergleichs der beiden Prüfsummen der Tippfehler sogar genau lokalisieren.

Aber Sie müssen sich natürlich nicht an diese Ideen halten. Nur ein paar Grundgedanken sollten Sie beibehalten:

— Das System sollte aus zwei Programmen oder -teilen bestehen: Eines für unsere Leser als Eingabehilfe und eines, mit dem wir die Listings fertig mit Prüfsummen versehen ausdrucken können. Der eigentliche Checksummer, den wir dann jeden Monat im Amiga-Magazin abdrucken werden, sollte so kurz wie möglich sein.

— Die maximale Größe der Prüfsumme muß ein gutes Mittelmaß darstellen. Eine zehnstellige Zahl ist sicherlich zuviel des Guten, wohingegen eine einstellige Prüfsumme natürlich keine große Sicherheit bieten kann.

— Ihr Checksummer sollte mit so vielen verschiedenen Programmiersprachen zusammenarbeiten wie nur irgend möglich. Wenn wir in jeder Ausgabe einen Checksummer für Basic-, einen für C- und einen für Assembler-Programme abdrucken müssen, so ist das sicherlich keine besonders gelungene Lösung des Problems.

— Machen Sie sich auch ein paar Gedanken, wie man Objectcodes, also Maschinenprogramme, möglichst platzsparend abdrucken könnte. Quelltexte und Basic-Listings sind nämlich ziemliche »Platzfresser« in Computer-Zeitschriften. Wenn man Programme in Form von Hexdumps veröffentlichen könnte, wäre dies sicherlich auch für den Leser von großem Vorteil, weil diese Hexdumps leichter und schneller einzutippen sind.

Gut wäre natürlich auch, gleich einen Packer beziehungsweise Entpacker für die oft sehr langen Objectcodes mit einzubauen.

Wir sind gespannt auf Ihre Ideen! Schicken Sie »Ihren« Checksummer an:

Markt & Technik Verlag AG
Redaktion Amiga-Magazin,
Aktion Checksummer
Hans-Pinsel-Str. 2
8013 Haar bei München

Und vergessen Sie nicht: Es kommt nicht darauf an, daß Sie so schnell wie möglich Ihren Checksummer einschicken. Eine ausgereifte und gelungene Lösung des Problems hat mehr Chancen auf den Preis von 1000 Mark. (tr)

Ein Tag in den Offenbacher Master Studios zu gewinnen!

Kennen Sie die Popgruppe »16 Bit« mit dem Song »Changing Minds«? Da hat es ein junger Elektrotechniker doch tatsächlich geschafft, mit Hilfe eines Amiga 500, ein paar Synthesizern und Effektgeräten einen Top-Hit zu landen.

Haben Sie schon einmal daran gedacht, ein Super-Musikstück für den Amiga selbst zu programmieren? Vielleicht ist dies hier Ihre große Chance, auch einer der großen Stars zu werden. Oder vielleicht werden Sie ja auch »nur« von einem Software-Haus als Starprogrammierer entdeckt (es sei hier nur an Chris Hülsbeck erinnert, der durch den ersten Preis beim Musikwettbewerb unserer Schwesterzeitschrift 64'er einen Vertrag als Programmierer bei Rainbow Arts erhielt). Bei diesem ersten Amiga-Musik-Programmierwettbewerb gibt es eine Menge zu gewinnen:

1. und 2. Preis: Eine Reise nach Frankfurt in die Offenbacher Master Studios. In diesem mit High-Tech vollgestopften Sound-Labor begann die Karriere des Michael Münzing und seinem Computer-Song »Changing Minds«. Ein Besuch bei Commodore in Frankfurt ist ebenfalls Bestandteil des Programms.

3. bis 5. Preis: Das Super-Musikprogramm »Sonix« von Aegis für Ihren Amiga.

6. bis 20. Preis: Eine Maxi-Single mit dem Hit »Changing Minds« der Gruppe »16 Bit«.

Was müssen Sie tun, um zu gewinnen? Schreiben Sie ein Musikstück für den Amiga. Komponieren Sie Ihren Anwärter auf die Top Ten. Gern gesehen (und gehört) werden natürlich Eigenkompositionen. Aber Sie können auch zum Beispiel Mozart, die Beatles oder Dire Straits in Bits und Bytes umsetzen. Ihrer Kreativität sind keine Grenzen gesetzt. Einzige Bedingungen: Das Musikstück muß ohne ein Editorprogramm lauffähig sein. Datenfiles für ein Musikprogramm wie zum Beispiel »Deluxe Music Construction Set« sind nicht zugelassen. Weiterhin darf der Song kein vollständig digitalisiertes Stück von Platte oder Kassette sein. Das soll natürlich nicht heißen, daß Sie überhaupt keine digitalisierten Sounds verwenden dürfen. Beweisen Sie also Ihr musikalisches Gespür und Harmonieempfinden.



Foto: Hilbig

Das ist Michael Münzing in voller Aktion. Mit seinem selbstkomponierten Song machte der Ex-Discjockey Karriere.

Schicken Sie Ihr Musikstück auf einer Diskette an:

Markt & Technik Verlag AG
Redaktion Amiga,
Stichwort
»Musikwettbewerb«
Hans-Pinsel-Str. 2
8013 Haar bei München
Einsendeschluß ist der
31.7.1987

Unter allen Einsendungen werden von einer ausgewählten Jury (Amiga-Redakteure, Mitarbeiter von Commodore und Michael Münzing von »16 Bit«) die besten Musikstücke ermittelt. Wir wünschen Ihnen viel Erfolg beim Programmieren! (tr)

Gestalten Sie Ihre Zeitschrift mit

Als Autor haben Sie die Gelegenheit, das Amiga-Magazin mitzugestalten. Ein Magazin von Lesern für Leser, so soll sich die Zeitschrift Ihnen präsentieren. Helfen Sie mit, dieses Ziel zu erreichen.

Sie kennen den Amiga? Beherrschen Sie eine Programmiersprache, oder können Sie gut mit kommerzieller Software umgehen? Dann haben Sie jetzt gute Chancen, Ihre eigenen Beiträge zum Amiga im Amiga-Magazin zu veröffentlichen. Das Amiga-Magazin soll eine Zeitschrift von Profis für solche, die es gerne werden möchten oder bereits sind, sein. Sie helfen mit Ihren fach- und sachkundigen Beiträgen anderen, sich mit der Komplexität dieses revolutionären Computers auszukennen. Ob Sie nun über Hardware oder die Programmierung in Assembler Bescheid wissen, was zählt ist Ihre Bereitschaft, das Amiga-Magazin abwechslungsreich, informativ und kreativ mitzugestalten.

Nicht nur als Vollprofi in Sachen Programmierung, sondern auch als geübter Anwender von bereits bewährter Software können Sie am Aufbau des Amiga-Magazins mitwirken. Zum Testen der neuesten Angebote auf dem Software-Markt sind schließlich keine Assemblerkenntnisse notwendig. Hier ist viel mehr Erfahrung im Umgang mit Dateiverwaltungen, Textverarbeitungs-Programmen, CAD- und Malprogrammen notwendig. Alle Anregungen und Artikelvorschläge, sowie Ihre Bereitschaft die Leser des Amiga-Magazins mit den aktuellsten Informationen zu versorgen, tragen dazu bei, diese Zeitschrift so optimal als möglich für Sie und die gesamte Leserschaft zu gestalten.

Alle veröffentlichten Beiträge werden natürlich entsprechend honoriert. Helfen Sie also mit, Sie können das Amiga-Magazin nach Ihren eigenen Wünschen formen. Und was gibt es Schöneres, als eine »eigene« Zeitschrift.

Sollten Sie Interesse an einer Mitarbeit beim Amiga-Magazin haben, dann richten Sie ein kurzes Anschreiben, auf dem Sie auch Ihr Fachgebiet umreißen, an folgende Anschrift:

Markt & Technik Verlag AG
z. H. Herrn R. Fieger
Stichwort: Amiga-Autor
Hans-Pinsel-Str. 2
8013 Haar bei München



AMIGA MAGAZIN

PROGRAMM-SERVICE

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzuschreiben, kann auf den bewährten Programm-Service zurückgreifen. Alle Programme, die mit dem Diskettensymbol ▪ im Inhaltsverzeichnis gekennzeichnet sind, gibt es auf Diskette. Lesen Sie aufmerksam die Anleitung in dem jeweiligen Artikel nach. Aus Aktualitätsgründen wird jeweils die abgedruckte Version angeboten.

Der detaillierte Disketteninhalt wird mit den Seitenzahlen in der nächsten Ausgabe abgedruckt.

Bei Fachfragen zu den Programmen wählen Sie bitte Telefon (089) 4613-640, bei Fragen zu Bestellung, Versand usw. Telefon (089) 4613-232 und bei Fragen zu Zeitschriften-Abonnements Telefon (089) 4613-362.

DAS TOP-LISTING DIESER AUSGABE:

Leckerbissen für C-Programmierer

Vergeblich sucht man in C nach Befehlen und Routinen zum Gestalten eigener Menüs. Um Ihnen die Arbeit mit den komfortablen Pull-down-Menüs in C zu ermöglichen, finden Sie in dieser Ausgabe alle C-Routinen, die benötigt werden, um menügesteuerte Programme zu erstellen. Ein kleines Demoprogramm unterstreicht die Leistungsfähigkeit der vorhandenen Routinen und gibt Ihnen gleichzeitig Einblick in die Arbeitsweise dieser Hilfsmittel.
Daneben finden Sie auf der Diskette auch alle anderen Programme, die im Inhaltsverzeichnis mit einem Diskettensymbol ▪ gekennzeichnet sind.

Best.-Nr. 48701 sFr 24,90/öS 299,-* **DM 29,90***

* inkl. MwSt. Unverbindliche Preisempfehlung

Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 4613-0. **Schweiz:** Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656. **Österreich:** Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0, Microcomputique E. Schiller, Fasangasse 24, A-1030 Wien, Telefon (0222) 785661, Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 833196. **Bestellungen aus anderen Ländern bitte nur schriftlich an:** Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die abgedruckte Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

Einlieferungsschein/Lastschriftzettel
(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)
bis 10 DM —— 90 Pf
über 10 DM (unbeschränkt) 1,50 DM
Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos
Auskunft hierüber erteilt jedes Postamt

Feld für postdienstliche Zwecke

Hinweis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.
1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Abkürzungen für die Ortsnamen der PGiroÄ:
Bln W = Berlin West, Kln = Köln, Dtmd = Dortmund, Lshfn = Ludwigshafen am Rhein, Esn = Essen, Ffm = Frankfurt am Main, Mchn = München, Nbg = Nürnberg, Hmb = Hamburg, Sbr = Saarbrücken, Han = Hannover, Stgt = Stuttgart, Krlh = Karlsruhe

Für Mitteilungen an den Empfänger

Bestellung Programm-Service — **Wichtig:** Lieferanschrift (Rückseite) nicht vergessen!

Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis
Summe bitte auf Vorderseite übertragen		Gesamtsumme:	

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

DM | Pf

für Postscheckkonto Nr.
14 199-803
Lieferanschrift und **Absender** der Zahlkarte

PLZ Ort

Verwendungszweck
M&T Buchverlag Programm-Service

Meine Kunden-Nr.:

DM | Pf | für Postscheckkonto Nr. **14 199-803**

Absender der Zahlkarte

PSchA | Postscheckkonto Nr. des Absenders | Postscheckteilnehmer

Zahlkarte/Postüberweisung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

DM | Pf | (DM-Betrag in Buchstaben wiederholen)

für Markt&Technik Verlag Aktiengesellschaft
in **8013 Haar**

Postscheckkonto Nr.
14 199-803
Postscheckamt
München

Ausstellungsdatum | Unterschrift

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

Einlieferungsschein/Lastschriftzette

DM | Pf

für Postscheckkonto Nr.
14 199-803
Postscheckam
Münche

für Markt&Technik Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in **8013 Haar**

Ihr Programm im Amiga-Magazin

Haben Sie bereits ein interessantes Programm für den Amiga geschrieben? Dann sollten Sie sich vielleicht überlegen, ob das von Ihnen entworfene Programm nicht für die Allgemeinheit der Amiga-Besitzer nützlich ist.

Nicht nur neue Spiele werden durch neue und leistungsfähige Computer wie den Amiga zu faszinierenden Meilensteinen in der Heimcomputer-Geschichte. Durch die steigende Speicherkapazität und die immer professionelleren Betriebssysteme können auch immer mehr höhere Programmiersprachen eingesetzt werden. Da die einzelnen Systeme heute immer komplexer aufgebaut sind, werden sogar die Betriebssysteme selbst in einer höheren Sprache erstellt. Auch hier ist der Amiga wieder ein sehr gutes Beispiel. Seine Seele, das Betriebssystem, wurde mit der modernen Sprache C programmiert.

Da für den Amiga in der Zwischenzeit ein bunt gemischtes Sammelsurium an verschiedensten Sprachen zur Verfügung steht, von Fortran bis Prolog, versorgen wir Sie natürlich mit brandheißen Informationen zu diesen Sprachen. Dabei wollen wir natürlich nicht an unseren Leser »vorbei programmieren«. Vielmehr haben Sie hier die Möglichkeit, Ihr Amiga-Magazin als Leser mitzugestalten. Wir suchen vor allem kurze Programme mit großer Wirkung, also Utilities oder wirkungsvolle Routinen, die jeder Leser und Programmierer sehr leicht für sich aus- und verwerten kann. Besonders nützlich ist auch eine genaue Dokumentierung des Quellcodes. Dadurch erschließt sich die von Ihnen verwendete Programmtechnik auch allen anderen Amiga-Anwendern. Die Programme werden leicht nachvollziehbar, und bestimmte Teile können so auf einfache Weise in eigene Programme eingebaut werden. So steigert sich ganz allmählich auch die Qualität der veröffentlichten Programme. Natürlich wird Ihr Beitrag bei Veröffentlichung entsprechend honoriert. Welche Sprache oder welchen Compiler Sie dabei verwenden, ist vollkommen egal. Ob Sie nun das bereits vorhandene Amiga-Basic benutzen oder sich mit Maschinensprache und C herumschlagen, wir werden Ihre Einsendung bewerten und, wenn für gut befunden, in einer der folgenden Ausgaben veröffentlichen. Wie gesagt, sie können jede Sprache verwenden, die für den Amiga erhältlich ist. Für diese Programme wird es dann eine eigene Rubrik geben, in der Sie die besten Lösungen gesammelt nach den einzelnen Sprachen geordnet wiederfinden.

Helfen Sie mit, Ihr Amiga-Magazin informativ und anregend zu gestalten. Sie haben die Möglichkeit dazu.

Wie gehen Sie bei der Einsendung vor? Ganz einfach, schicken Sie uns Ihr Programm mit genauer Beschreibung und Diskette, sowie der Angabe der verwendeten Sprache an folgende Adresse:

Markt & Technik Verlag AG
Amiga-Redaktion
z. H. Herrn R. Fieger
Stichwort:
Amiga-Programmierung
Hans-Pinsel-Str. 2c
8013 Haar bei München

Gesucht: Tips & Tricks zum Amiga

Sie sind ein begeisterter Amiga-Fan? Sie kennen Ihren Computer inzwischen so gut, daß die Begriffe »CLI«, »Amiga-DOS« oder »Workbench« für Sie keine Fremdwörter mehr darstellen? Dann beteiligen Sie sich doch an der Tips & Tricks-Rubrik »Ihres« neuen Magazins!

Die Tips & Tricks-Ecke im Amiga-Magazin ist eine reine »von-Lesern-für-Leser«-Rubrik. Das bedeutet, daß Sie durch Ihre Beiträge und durch Ihre guten Ideen anderen Lesern helfen können, mit dem neuen Computer besser zurechtzukommen. Sicherlich erinnern Sie sich noch an die Zeit, wo Sie begierig jede Zeile Information zum Amiga mit Heißhunger verschlangen. Gerade als Einsteiger ist man auf die Erfahrung und das Wissen von »Profis« angewiesen.

Sie haben eine neue Idee, wie man mit einem kleinen Amiga-DOS-Programm die Arbeit mit dem Computer vereinfachen kann? Einschicken!

Sie haben ein raffiniertes, kurzes Basic-, C- oder Assembler-Programm ausgeklügelt? Einschicken!

Sie haben einen Trick herausgefunden, wie man mit der einen oder anderen Software komfortabler arbeiten kann? Einschicken!

Sie haben Ihren Amiga durch eine kleine Bastelei aufgewertet? Einschicken! Jede gute Idee wird in »Ihrer« Rubrik veröffentlicht. Natürlich bekommen Sie dann von uns ein entsprechendes Honorar. Wichtig ist nur, daß Sie in einer kurzen Beschreibung den Grundgedanken Ihrer Idee zu Papier bringen. Wenn Sie ein Programm einschicken, sollten Sie nicht vergessen, eine Diskette mitzuschicken. Und sollte einmal ein Beitrag, aus welchen Gründen auch immer, nicht zur Veröffentlichung geeignet sein, bekommen Sie Ihre gesamten Unterlagen natürlich wieder zurückgesandt.

Schreiben Sie an:
Markt & Technik Verlag AG
Redaktion Amiga-Magazin,
Aktion Tips & Tricks
Hans-Pinsel-Str. 2
8013 Haar bei München

Eines sollten Sie nicht vergessen: Sie müssen kein Vollprofi sein, um bei dieser Aktion mitzumachen. Auch Tricks, die für Sie schon längst zur Routine geworden sind, könnten anderen Lesern eine wichtige Hilfe sein. Vor allem kurze und leicht durchschaubare Listings sind begehrt. Wir wünschen Ihnen viel Spaß beim Programmieren und Knobeln! (tr)

Vorschau 8/9 1987



Was bringt die Software?

Kein Computer ohne Software! An dieser Binsenweisheit orientieren sich nicht nur die Computer-Freaks aus aller Welt, sondern natürlich auch die Amiga-Redaktion.

Wie schon in dieser Ausgabe, so werden ebenfalls im zweiten Amiga-Magazin wichtige und interessante Neuerscheinungen auf dem Software-Markt exklusiv von uns für Sie getestet. So behalten Sie die Übersicht und bekommen außerdem wichtige Tips und Hilfen für eine eventuelle Kaufentscheidung.

Von Grafik und Musik über Spiele bis hin zu Datei- und Textprogrammen wird keine wichtige Software-Sparte ausgelassen werden. Unter anderem berichten wir natürlich wieder ausführlich über das Programmieren mit und auf dem Amiga. Der Umgang mit dem CLI und mit Programmiersprachen wird erläutert, einige gute Beispiel-Listings runden das Ganze entsprechend ab. Hobbyprogrammierer wie auch Fortgeschrittene werden im nächsten Amiga-Magazin sicherlich fündig werden.



Kartenspiele

Der Amiga 2000 soll für Furore sorgen; und das nicht nur im 68000- sondern dank seiner Steckkarten auch im MS-DOS-Bereich. Was die Karten leisten und wie kompatibel sie sind, werden wir in ausführlichen Tests ergründen.

CES

In der nächsten Ausgabe finden Sie einen Messebericht von der »Consumer Electronics Show«. Die heißesten Entwicklungen auf dem Soft- und Hardware-Markt kommen hier zum ersten Mal an das Licht der Öffentlichkeit.

Inserenten

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier (aa)
Leitender Redakteur: Georg Klinge (gk)
Redaktion: René Beaupoil (rb), Achim Hübner (ah), Jörg Kähler (jk), Gregor Neumann (gn), Thomas Röder (tr), Norbert Jungmann (nj), Roland Fieger (rf), Dieter Mayer (dm), Boris Schneider (bs), Peter Pfliegensdörfer (pd), Karsten Schramm (ks), Gerd Donaubauer (do)

Redaktionsassistenz: Monika Lewandowski (222), Andrea Kaltenhauser (202), Bärbel Pasternok (202)
Fotografie: Janos Feitser/Jens Jancke, Hilbig, Laserland

Layout: Leo Eder (Ltg.), Rolf Raß (Cheflayouter), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 56 56, Telex: 862329 mut ch
USA: M & T Publishing, Inc: 501 Galveston Drive, Redwood City, CA 94063; Tel. (415) 366-3600, Telex 752-351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleitung: Klaus Buck

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenleitung: Alicia Clees (313)

Anzeigenverwaltung und Disposition: Patricia Schiede (172), Monika Stoiber (147)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Vertriebsleiter: Helmut Grünfeldt (189)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (07 11) 64 83-0

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement kann jederzeit zum Ende des bezahlten Zeitraums gekündigt werden.

Bezugspreise: Das Einzelheft kostet DM 7,—. Der Abonnementspreis beträgt im Inland DM 79,— pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,— für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z. B. USA) um DM 38,—, in Ländergruppe 2 (z. B. Hongkong) um DM 50,—, in Ländergruppe 3 (z. B. Australien) um DM 68,—.

Druck: R. Oldenbourg, Hürderstr. 4, 8011 Kirchheim

Verantwortlich: Für redaktionellen Teil: Albert Absmeier. Für Anzeigen: Alicia Clees.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

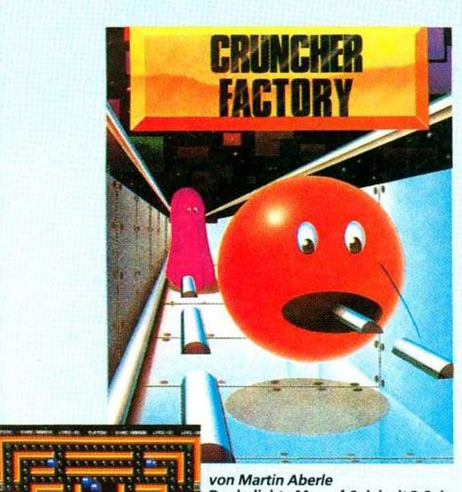
Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Aktionäre, die mehr als 25% des Kapitals haben: Otmar Weber, Ingenieur, München, Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr

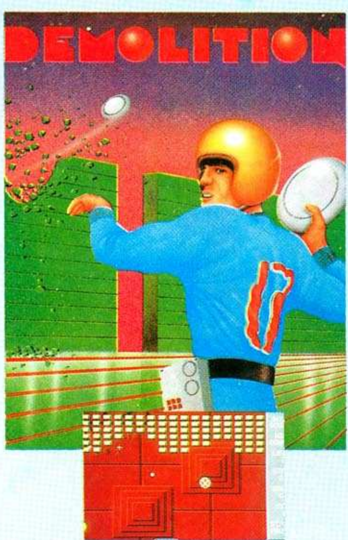
Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

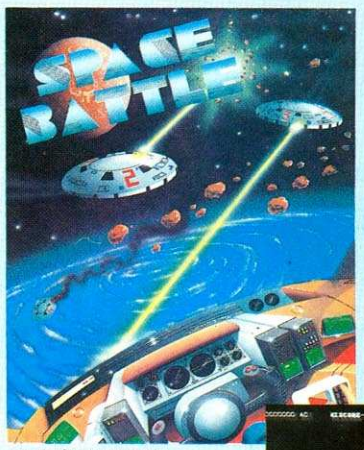
SPITZEN-SOFTWARE FÜR AMIGA ZUM TASCHENGELD-PREIS!



von Martin Aberle
Das beliebte Mampf-Spiel mit 2-Spieler-Option (gleichzeitig!) und Level-Editor (eigene Bilder erstellen). „Vor allem zu zweit macht Cruncher Factory Spaß und angesichts des sehr attraktiven Preises kann man das Spiel ...empfehlen." (HAPPY COMPUTER)



von Martin Aberle
Brechen Sie die Mauer ein, aber achten Sie auf diverse Monster und Ufos, die durch die Gegend fliegen. Die aktuelle Version des Spielhallenklassikers.



von Andreas Hommel
Das klassische Weltraumspiel in einer hervorragenden Neuauflage. Natürlich mit Thrust und Hyperspace sowie einer 2-Spieler-Option (gleichzeitig, gegeneinander oder zusammen).



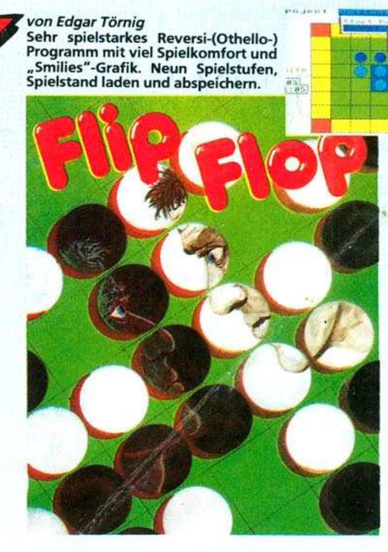
von Andreas von Lepel
Jagen Sie in diesem spannenden Weltraumrennen zu zweit durch eine futuristische Landschaft. Tolle Grafik im „Dual-Playfield-Modus" (AMIGA-Spezialität).

CHALLENGER



von Edgar Törnig
Sehr spielstarkes Reversi-(Othello-) Programm mit viel Spielkomfort und „Smilies"-Grafik. Neun Spielstufen, Spielstand laden und abspeichern.





© SE★DESIGN

PHALANX

von Oliver Czesla
Ein Weltraumspiel der Extraklasse mit einer Großinvasion von Sprites, die sich in immer neuen Formationen auf Ihr Raumschiff stürzen. Dabei wird als Besonderheit der gesamte PAL-Bildschirm (256 Zeilen) als Spielfläche ausgenutzt.



für fast alle Computer-Typen, die gegen erstklassige Bezahlung Spiele von internationalem Niveau schreiben. Wenn Sie Interesse haben, wenden Sie sich noch heute an uns.

Und das ist ist erst der Anfang! Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA - am besten kostenlos den aktuellen Katalog anfordern.
Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung, starker Grafik, Highscore-Liste und Digisound. KINGSOFT-Spiele gibt's überall da, wo es gute Software gibt.





MADE IN GERMANY

F. Schäfer · Schnackebusch 4 · 5106 Roetgen
☎ 02408/5119 (nicht aufgeben!) Fax 02408-5213

Klasse statt Masse

GO AMIGA! Datei



Das sind die Vorteile von GO AMIGA Datei

- Pull-Down Menüs, Maus- und/oder Tastatursteuerung.
- Arbeiten mit internem und externem Speicher (RAM), kompatibel zu Harddisk.
- Schnittstelle zu anderen Programmen mittels ASCII-Datei.
- Leistungsfähiges Such- und Sortierverfahren.
- Editieren der Bildschirmmasken während der Arbeit.
- Vielseitige, komfortable Druckersteuerung von Etiketten bei Listenausdruck.
- Mehrere, individuelle Bildschirmmasken für ein und dieselbe Datei.
- Frei wählbare Darstellung von Zahlen, internationalen Währungen (z. B.: DM, sFr., ÖS).
- Mail Merge Funktion.
- Sicheres Abspeichern, auch wenn Disk voll oder schreibgeschützt ist.
- Diaschau — verwaltet und zeigt Grafikbilder am Bildschirm.
- Tonschau — verwaltet und spielt digitalisierte Töne.
- Filmschau — Dia- und Tonschau gleichzeitig.
- Ausführliches deutsches Handbuch.

Der Preis wird Sie erstaunen! GO AMIGA Datei kostet DM 199,— oder sFr. 178,— oder ÖS 1890,—.

Bestellservice:

BRD: 0041-1-3115959
CH: 01-3115959

Geschäftszeiten:
10.00–12.30, 13.30–18.30 Uhr, außer montags, Sa.: 10.00–16.00 Uhr.

Versand ins Ausland nur Vorkasse (Scheck, bar) zzgl. DM 6,– Porto. Händleranfragen erwünscht.

softwareland
Franklinstraße 27
CH-8050 Zürich (Schweiz)